

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра інформаційних технологій та комп'ютерної інженерії

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня *бакалавра*

Студента Галушка Олександра Геннадійовча

академічної групи 126 – 18 – 1

спеціальності 126 «Інформаційні системи та технології»

на тему: “Розробка хмарного хостінгу на основі програмного забезпечення з відкритим кодом”

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		Рейтинговою	Інституційною	
кваліфікаційної роботи	<i>к.т.н., доц. Соколова Н.О.</i>			
розділів:				
Рецензент	<i>к.т.н. Гуліна І.Г.</i>			
Нормоконтролер	<i>д.т.н., проф. Коротенко Г.М.</i>			

Дніпро
2022

ЗАТВЕРДЖЕНО:
завідувач кафедри
Інформаційних технологій та комп'ютерної інженерії

(повна назва)

_____ д.т.н., проф. Гнатушенко В.В.
(підпис) (прізвище, ініціали)

«_____» _____ 2022 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра

студенту Галушку О.Г. академічної групи 126-18-1
спеціальності: 126 «Інформаційні системи та технології»
на тему «Розробка хмарного хостінгу на основі програмного
забезпечення з відкритим кодом»

затверджену наказом ректора НТУ «Дніпровська
політехніка» від 18.05.2022 р. №268-с

Розділ	Зміст	Терміни виконання
1. Розділ 1	Проаналізувати стан області рішення задач	02.05.2022 – 08.05.2022
	Провести огляд програмного забезпечення з відкритим кодом	09.05.2022 – 16.05.2022
2. Розділ 2	На основі відповідних матеріалів проаналізувати варіанти оптимального рішення для розробки власного проекту	17.05.2022 – 05.06.2022

Завдання видано _____ доц. Соколова Н.О.
(підпис) (прізвище, ініціали)

Дата видачі: 02.05.2022 р.

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: с. 55, рис. 45, додатків 1, джерела 4.

Об'єкт розроблення: хмарний хостінг на основі виділеного серверу.

Мета дипломного проекту: розробка та конфігурування хмарного хостингу на основі програмного забезпечення з відкритим кодом.

Предмет дослідження – особливості конфігурування виділеного серверу на основі програмного забезпечення з відкритим кодом для розробки хмарного хостингу.

У вступі подано стан проблеми, здійснено аналіз аналогів рішення, сформульовані задачі, які потрібно виконати для успішного виконання.

В першому розділі наведені основні відомості про існуючі види хостингу, проведено аналіз програмного забезпечення з відкритим кодом, сформовано набір необхідного програмного забезпечення для роботи серверу.

У другому розділі наведена проектна складова вирішення завдання, щодо розробки хмарного хостингу та його реалізація як проекту; окрім того проведена перевірка працездатності на прикладі розміщення сайту.

Практичне значення проекту полягає у конфігурації сервера та додатків з відкритим кодом, відповідно до поставлених задач

Розроблене технічне рішення може бути впроваджено для комерційного використання або для розміщення власного сайту (наприклад портфоліо).

ХМАРНИЙ ХОСТИНГ, ВИДІЛЕНИЙ СЕРВЕР, ВЕБ-ХОСТИНГ, САЙТ, LINUX, ВІРТУАЛЬНИЙ СЕРВЕР, КОНФІГУРАЦІЯ.

ABSTRACT

Explanatory note: pp. 55, Fig. 45, Appendices 1, sources 4.

Development object: cloud hosting based on a dedicated server.

The purpose of the diploma project: development and configuration of cloud hosting based on open source software.

The introduction presents the state of the problem, analyzes the analogues, formulates the tasks that need to be performed for successful implementation.

The subject of research - the features of configuring a dedicated server based on open source software for the development of cloud hosting.

The first section provides basic information about existing types of hosting, analysis of open source software, formed a set of necessary software for the server.

The second section presents the project component of solving the problem of developing cloud hosting and its implementation as a project; in addition, a health check was performed on the example of site placement.

The practical significance of the project lies in the configuration of the server and open source applications, in accordance with the objectives

The developed technical solution can be implemented for commercial use or for hosting your own site (eg portfolio).

CLOUD HOSTING, DEDICATED SERVER, WEB HOSTING, WEBSITE, LINUX, VIRTUAL SERVER, CONFIGURATION.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. Аналізу стану області рішення задачі.....	8
1.1 Визначення хмарного хостингу.....	8
1.2 Визначення веб-хостингу.....	8
1.3 Види хостингу.....	10
1.4 Огляд програмного забезпечення для віртуального виділеного серверу...	13
1.4.1 Визначення веб-серверу.....	13
1.4.2 Порівняльна характеристика веб-серверів Apache та Nginx.....	13
1.4.3 Порівняльна характеристика СУБД MariaDB та MySQL.....	15
1.4.4 Характеристика PHP.....	15
1.5. Характеристика протоколу SSH.....	16
1.6 Характеристика FTP та SFTP.....	17
1.7. Характеристика роботи фаєрволу.....	19
1.8 Висновки до першого розділу.....	20
РОЗДІЛ 2. Розробка хмарного хостінгу на основі програмного забезпечення з відкритим кодом.....	22
2.1 Первинне налаштування виділеного серверу.....	23
2.2 Встановлення та конфігурування веб-серверу Apache.....	26
2.3 Інсталювання PHP.....	32

2.4 Інсталювання і конфігурація СУБД MariaDB.....	34
2.5 Встановлення системи керування вмістом WordPress.....	37
2.6 Налаштування протоколу ftp.....	43
2.7 Висновки до другого розділу.....	52
ВИСНОВКИ.....	52
ВИКОРИСТАНІ ДЖЕРЕЛА.....	53
Додаток А.....	55

ВСТУП

В наш час люди оцінили зручність мережі Інтернет. Вона дозволяє нам вести бізнес, дізнаватись нову інформацію, бронювати номери готелі та врешті решт розважатись. Можливості, котрі надає мережа Інтернет своїми ресурсами безмежна.

Наявність свого сайту або веб-додатку підвищує шанси на успіх компанії та створює її імідж в мережі Інтернет. Якщо компанія не має хоча б простого сайту-візитки з інформацією про неї, то ця компанія в очах клієнтів виглядає не серйозно або застарілою.

Декілька років тому, щоб дізнатись подробиці про роботу або здійснити замовлення у підприємства, потрібно було знайти номер телефону для зв'язку або приїхати до офісу і зробити це тільки у робочий час.

Сайт сьогодні – це віртуальний офіс компанії, який працює безперервно і самостійно. Кожен клієнт може переглянути необхідну інформацію або здійснити замовлення в любий час доби [1].

Створення та розміщення сайту в мережі – це можливість заявити про себе на весь світ. За допомогою нього можна сповістити потенційних клієнтів, постачальників та партнерів, про необхідну інформацію, розповідаючи про свої переваги та умови, котрі відрізняють вас від конкурентів.

Однією з умов успішності сайту розміщеного в мережі – являється його веб-хостинг. Саме ця складова відповідає за стабільну і безперервну роботу «онлайн-офісу», швидкість донесення інформації до кінцевого користувача, безпеку персональних даних та багато іншого.

Метою роботи є запропонувати рішення для побудови приватного веб-хостингу за допомогою інструментів з відкритим кодом, який надаватиме

можливість гнучкого налаштування , завантаження контенту майбутнього веб-сайту та його стабільну роботу в мережі Інтернет.

Об'єкт дослідження – розробка хмарного хостингу для розміщення сайту в мережі Інтернет.

Предмет дослідження – особливості конфігурування виділеного серверу на основі програмного забезпечення з відкритим кодом для розробки хмарного хостингу.

Для досягнення поставленої цілі були встановлені наступні задачі:

1. Провести аналіз можливих рішень та варіантів хостингів для розміщення сайту.
2. Проаналізувати існуюче програмне забезпечення з відкритим кодом для реалізації та управління веб-хостингом.
3. Організація управління сервером за допомогою протоколу SSH .
4. Можливість завантаження контенту на сервер по протоколу S/FTP.
5. Встановлення брандмауеру для безпеки.

РОЗДІЛ 1

АНАЛІЗУ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ

1.1. Визначення хмарного хостингу

Хмарний хостинг – це тип веб-хостингу, який працює за моделлю розміщення на хмарному сервері, тобто віртуальному виділеному сервері. Дані веб-сайту при використанні веб-хостингу, розміщені на віртуальному або фізичному сервері котрий називається виділеним. Дані на хмарному хостингу поширюються на декілька серверів.

Однією за переваг даного виду хостингу є те, що при надмірному навантаженні одного з серверів або виходу з ладу, інші сервери приймають на себе навантаження. Тим самим зрівнюючи навантаження і корегуючи роботу ресурсів. Таким чином це робить хмарний хостинг більш надійним, відмовостійким та безпечним для розміщення на ньому веб-проектів.

Доречність використання хмарного хостингу зазвичай визначається амбіціями та ресурсоемністю веб-проекту. Наприклад, Google розподіляє свої ресурси на сотні серверів у хмарі. Це сприяє надійності в роботі та стабільністю у використанні [1].

1.2. Визначення веб-хостінгу

Веб-хостинг — це послуга для розміщення веб-сайтів або веб-додатків на серверах та надання доступу до них безперебійного доступу через мережу Інтернет. Підприємства, що надають такі послуги називаються хостинг - провайдерами. Вони надають у використання ресурси власних або

орендованих серверів, які частіше всього знаходяться по всьому світу в дата-центрах[3].

Існує декілька видів хостингу ,які відрізняються в залежності від основної цілі або необхідних можливостей веб-проекту, який буде розміщено. Таким чином найпростішим видом являється хостинг для статичних сайтів. Частіше всього це сайти, які використовуються для сайтів-візиток, односторінкових сайтів або сайтів, у яких не передбачається взаємодії з користувачем. Тож часто подібні сайти не використовують бази даних та PHP модулі. Хостинг для статичних сайтів надається за мінімальною ціною або безплатно, на умовах розміщення реклами від хостинг-провайдера, тому що вони не спричиняють великого навантаження на сервери хостинг-провайдера.

Для повноцінних комерційних сайтів або веб-додатків більшої складності використовуються вже більш дорогі варіанти хостингу з можливістю використання баз даних, платформ розробки (наприклад PHP, Python, Java та інших) або панелей управління для керування сервером.

Частіше всього хостинг-провайдер надає можливість використання власної або однієї з популярних панелей управління хостингом (наприклад cPanel, DirectAdmin, ISPmanager та інші). Саме за допомогою таких панелей управління, клієнт має змогу розміщувати, налаштовувати та діагностувати необхідні частини сайту, використовуючи графічний інтерфейс.

Невід'ємною частиною хостингу являється його аптайм (відсоток часу його перебування в робочому стані) і робота технічної підтримки. Адже це забезпечує безперебійну роботу та виправлення складнощів у роботі сайтів та веб-додатків, котрі розміщені на серверах хостинг-провайдера.

1.3. Види хостингу

Кожен вид хостингу передбачає розміщення на ньому веб-проектів, яким потрібні обов'язкові додатки для роботи без надлишкових функцій. Додаткові функції хостингу впливають на ціноутворення тарифу за яким буде оплачуватись хостинг (найчастіше мінімальна оренда від 1 місяця). Наглядна схема видів хостингів зображена на рисунку 1.1.

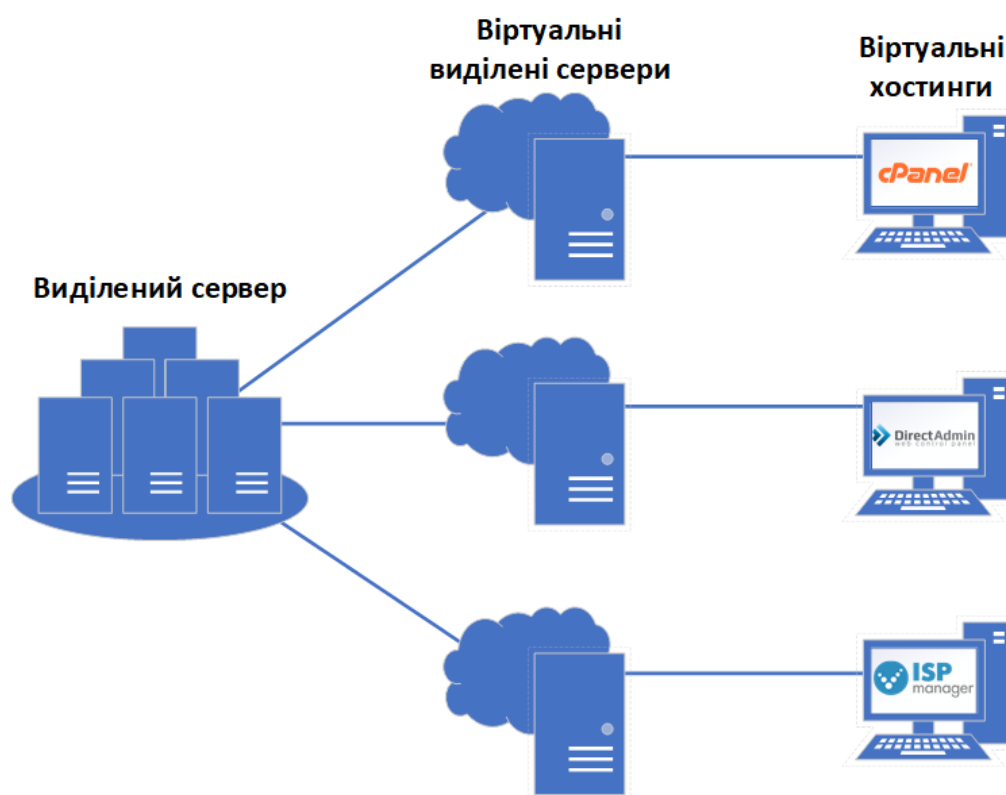


Рисунок 1.1 - Схема видів веб-хостингів

Безкоштовний віртуальний хостинг:

Надається хостинг із обмеженим функціоналом порівняно з платними тарифами хостингу. Обслуговування подібних тарифів компенсується за

рахунок розміщення реклами хостинг-провайдера на сайті для залучення нових клієнтів.

Віртуальний хостинг:

Для кожного віртуального хостингу на сервері зарезервованій певний ліміт використання ресурсів. Також віртуальні хостинги, які знаходяться на одному сервері, використовують набір програмного забезпечення, який передбачено сервером. Так «сусідні» віртуальні хостинги використовують ідентичні версії програмного забезпечення. Веб-сайт розміщено на одному сервері, одночасно з іншими клієнтами і використовують загальну IP адресу.

Одним з мінусів може бути те, що при сильному навантаженні одного з віртуального хостингу на сервері (наприклад DDoS атака), можливе погіршення роботи інших сайтів, котрі знаходяться на сервері.

Віртуальний виділений сервер (VDS):

Один з видів хостингу, в рамках якого, за умовами користування надають орендатору віртуальний виділений сервер. Це повноцінний аналог фізичного виділеного серверу з великою кількістю переваг, високою стабільністю, гнучкістю в управлінні та налаштуванні, відмовостійкістю, кількістю розташованих поруч «сусідів» на сервері та набагато меншими фінансовими витратами .

Таким чином користувач має повноцінний сервер з встановленою операційною системою (наприклад Centos, Ubuntu, Debian, FreeBSD, Windows та інші), та можливістю практично повного управління.

При використанні віртуального серверу можна створювати власні версії системних бібліотек або замінювати існуючі. Орендатор VDS (Virtual Dedicated Server) має право видаляти, завантажувати, змінювати будь-які

файли, включаючи системні файли або директорії, а також власноруч завантажувати, налаштовувати чи видаляти будь-яке доступне програмне забезпечення [4].

Однією з вагомих переваг VDS – є гнучкість налаштування та повне управління системою для використання у цілях веб-проектів.

Виділений сервер

Вид хостингу, при якому покупцю надається окремий фізичний сервер у повне управління та обслуговування (на відміну від віртуального виділеного серверу). Орендатор має повний контроль над сервером, у тому числі вибір операційної системи, можливість апгрейду технічних частин, обладнання та інше [5].

Як правило, адміністрування фізичних серверів надається хостинг-компанією як додатковий платний додаток. Так як частіше за все, сервери знаходяться у різних кутках світу в дата-центрах.

Оренда фізичного серверу оправдовує себе незалежністю від «сусідів», так як в дата-центрах кожен сервер не залежить один від одного, має додаткові джерела підключення до мережі, системи вентиляції та не піддає ризику в роботі для інших машин навантаженням. Це в деякій мірі може підвищити відмовостійкість, запропонувати зменшення витрат та більший прибуток від інвестицій.

В нашому випадку було обрано саме віртуальний виділений сервер, адже він являється бюджетним рішенням для створення гнучкої, стабільної і продуктивної роботи веб-хостингу.

1.4. Огляд програмного забезпечення для віртуального виділеного серверу.

1.4.1. Визначення веб-серверу.

Веб-сервер це комп'ютер розміщений в мережі з певним програмним забезпеченням. На серверах даного типу в мережі зберігається безліч інформації. Усі веб-додатки та веб-сторінки зберігаються на веб-серверах, які приймають запит від користувача або звернення до певного додатку, а на виході користувач отримує результат обробки веб-сервером.

Одним з найпопулярнішого серверного програмного забезпечення, являється Apache і Nginx [6]. Згідно з сервісом SimilarTech, Nginx являється більш популярним і в більшості випадків використовується на UNIX-подібних системах (як і Apache), хоч і має можливість використання його на операційних системах Windows. На даний час Apache використовують близько 7 мільйонів веб-сайтів, в той час як Nginx використовують близько 7,4 мільйони веб-сайтів [7].

1.4.2. Порівняльна характеристика веб-серверів Apache та Nginx.

У Apache та Nginx використовуються різні підходи до обробки інформації та взаємодії з користувачем. Одним з пунктів являється метод обробки підключень з клієнтами. Apache на кожен запит користувача створює окремий процес. Це виглядає наступним чином – клієнт робить запит до серверу, веб-сервер створює окремий процес на цей запит, відповідає клієнту і блокує процес до тих пір, поки клієнт не розірве з'єднання. Даний метод потребує більших ресурсів серверу у випадку, якщо кількість клієнтів або запитів буде великою.

Так як в операційних системах кожен процес потребує повної кількості ресурсів, то у випадку накопичення запитів, буде підвищуватись

навантаження на оперативну пам'ять та процесор. Для невеликих проектів дане програмне забезпечення являється збалансованим рішенням між складністю налаштування і продуктивністю.

Обробка запитів у Nginx складається з майстер-процесу та декількох дочірних процесів. Майстер-процес зазвичай один – він створює дочірні процеси, зчитує конфігурацію і відкриває порти. Дочірні процеси обслуговують усі з'єднання з клієнтами і не блокують їх. У Nginx використовується нескінченний цикл, котрий проходиться по всім з'єднанням і відповідає на запити клієнтів. Коли з'єднання закривається, то воно відразу ж видаляється з робочого циклу. Дане рішення переважно підходить для проектів, які обслуговують тисячі запитів одночасно (наприклад сайт нашого університету використовує nginx). При цьому, навантаження на процесор і використання пам'яті частіше рівномірні та без явних піків, котрі можна відслідковувати в програмах моніторингу навантаження систем.

У порівнянні з віддачею контенту, Apache може генерувати динамічний і статичний контент. Він являється готовим рішенням на відміну від Nginx, якому потрібне налаштування проксування та додаткового інструменту для генерації динаміки.

Однією з головних переваг Apache, являється зручне конфігурування обробки з'єднань на рівні директорій. Це робиться за допомогою файлу `.htaccess`, який дає змогу налаштувати кешування, перенаправлення запитів, блокування по IP та багато інше.

Nginx не підтримує налаштування на рівні каталогів. Для нього назначений конфігураційний файл на весь проект.

В нашому випадку було обрано Apache, тому що планується підготовка серверу для розміщення сайту з невеликим навантаженням з'єднань та наявність зручного керування обробки з'єднань на рівні директорій.

1.4.3. Порівняльна характеристика СУБД MariaDB та MySQL

Однією з складових нашого серверу являється база даних. MariaDB знаходиться у статусі постійної розробки, швидкого оновлення, так як проект являється відкритим проектом.

Починаючи з версії 10.1 MariaDB дає змогу шифрування даних на рівні сховища. Це відноситься як повних табличних просторів, так і окремих таблиць. На відміну від MySQL, в MariaDB запитуються тільки ті таблиці, які необхідні для виконання запиту [8].

Отже, MariaDB являється автономною системою управління базами даних, завдяки безперервному розвитку і вже отримала перевагу над MySQL. Вона використовується навіть такими проектами як: Google, Mozilla, OpenBSD, openSUSE, Fedora та інші., тому була обрана для побудови нашого проекту.

1.4.4. Характеристика PHP

На завершення основного програмного забезпечення нашого серверу, було обрано встановлення модулю PHP 7, який потребується для більшості CMS(Content Management System) веб-сайтів (наприклад WordPress, ModX, Joomla, Drupal та інші) [10].

PHP являється популярною мовою програмування з відкритим кодом, особливо у середовищі веб-розробників. Мова програмування PHP зручна для написання скриптів та повноцінних веб-додатків, котрі виконуються на стороні серверу. Це найпопулярніша сфера використання, адже ця мова програмування була розроблена саме для веб-розробок. Для повноцінної роботи веб-додатку або сайту розробленого на PHP, необхідні: сервер, парсер та клієнтське програмне забезпечення (наприклад веб-браузер), яке буде відображати результат обробки коду.

Основними перевагами PHP є простий та інтуїтивно зрозумілий інтерфейс, кросплатформність, гнучкість та масштабованість.

1.5. Характеристика протоколу SSH

Secure Shell (SSH) – захищений протокол віддаленого адміністрування. Був розроблений для здійснення віддаленого управління операційними системами і тунелюванням TCP-з'єднання.

Зазвичай управління комп'ютером відбувається віддалено. Наприклад, якщо використовується виділений віртуальний сервер. Для виконання команд на ньому і використовується протокол SSH.

SSH це протокол, використовує клієнто-серверну модель для аутентифікації віддалених систем та забезпечення шифрування даних, та обмін при наданні віддаленого доступу.

За замовчуванням при роботі використовується 22 порт TCP, але може бути змінений за необхідності. Для створення підключення по SSH клієнт повинен ініціювати підключення з сервером, забезпечивши захищене з'єднання і підвердивши свій ідентифікатор (ключ). Схеми роботи підключення зображені на рисунку 1.2.

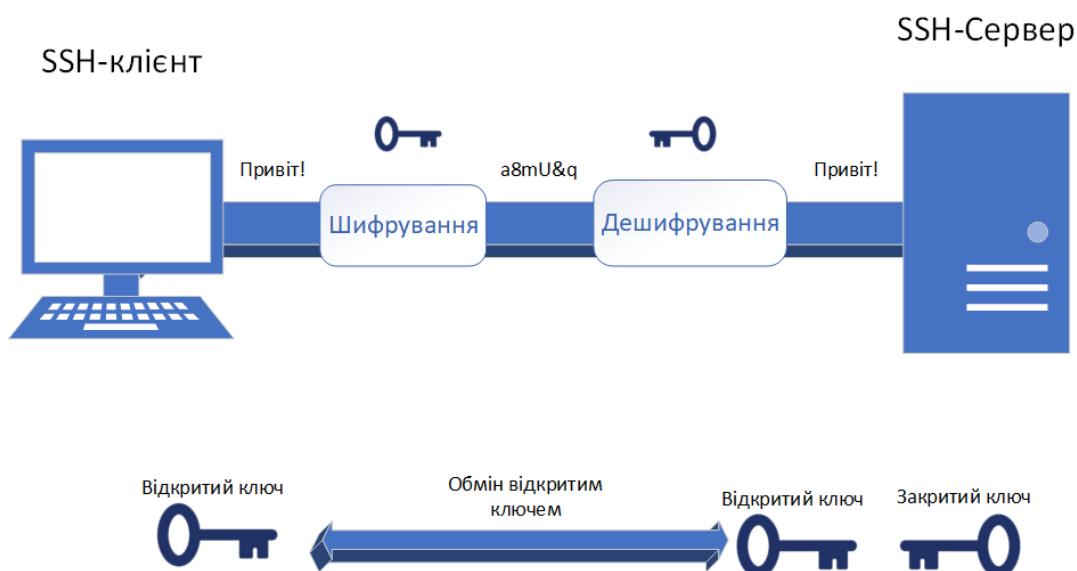


Рисунок 1.2 – Схема роботи SSH

Під час підключення користувач має змогу проводити команди, як на своєму пристрої. Часто використовують перелік команд для навігації в системі, оновлення додатків, встановлення з резервної копії бази даних, виконання скриптів на інше.

1.6. Характеристика FTP та SFTP

Протокол мережевого зв'язку FTP надає можливість передачі даних між декількома машинами або серверами. FTP-клієнти є стандартними для більшості операційних систем, існуючи з графічним інтерфейсом або без нього.

FTP створює два логічних зв'язки між клієнтом та сервером – канал контролю та канал даних. Запускаючи канал контролю з сервером перед початком сеансу, клієнт встановлює з'єднання. Для цього використовується 21

порт. На другому етапі підключення для передачі даних використовується з'єднання через 20 порт. Клієнт передає команди через канал контролю, в той час як обмін даними проходить через з'єднання передачі даних [11].

Зазвичай для підключення створюється ftp-користувач з власними даними для підключення (логін та пароль). Це дозволяє встановити особливі привілеї, доступи та квоти для користувачів. Схема роботи ftp продемонстрована на рисунку 1.3.

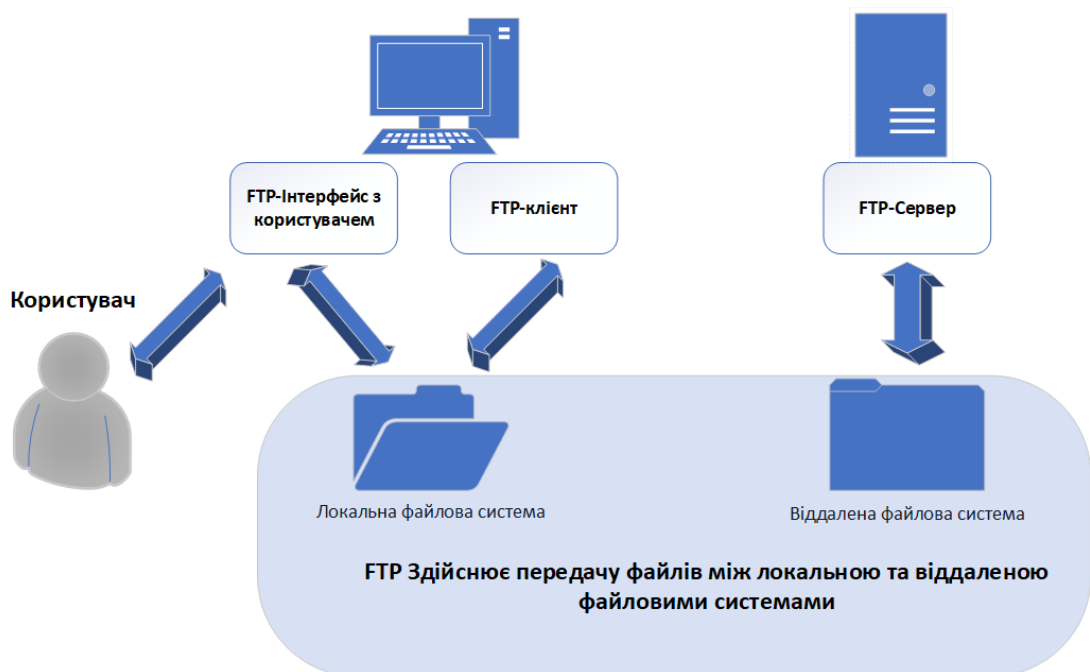


Рисунок 1.3 – Демонстрація роботи FTP

SFTP (SSH File Transfer Protocol) – це протокол передачі файлів, який працює поверх захищеного каналу. Являється аналогом FTP протоколу з підвищеним рівнем захисту, який використовується у протоколі SSH. Рекомендується використання саме протоколу SFTP за для цілісності і безпеки даних від перехоплення третьою стороною. Наглядна схема роботи протоколів передачі даних приведена на рисунку 1.4.

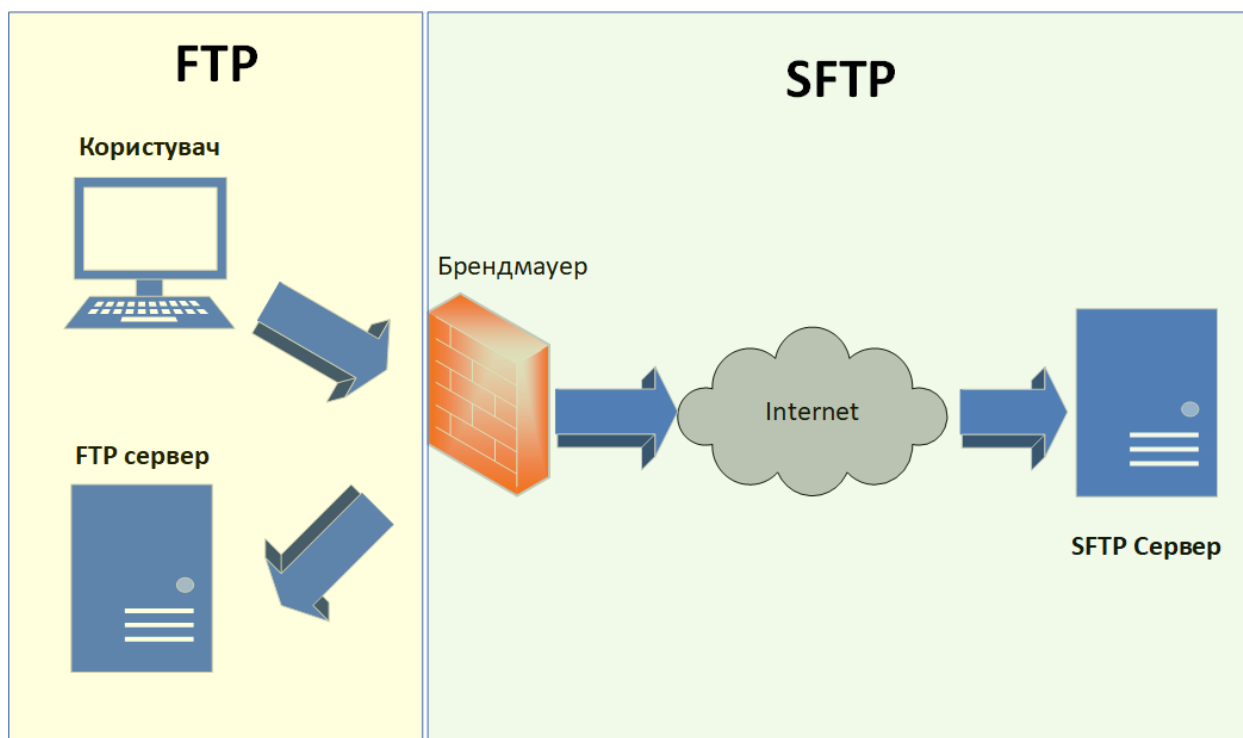


Рисунок 1.4 – Схема роботи протоколів FTP та SFTP

1.7. Характеристика роботи фаєрволу

Фаєрволом (або брендмауер) називають фізичний або програмний застосунок, сконфігурований для допуску, відмови, шифрування або пропуску мережевого трафіку між областями різних мереж за бажаним набором правил встановлених адміністратором [12].

Сервери постійно піддаються різноманітним атакам або скануються злочинцями у пошуках вразливостей. Як тільки злочинець знаходить вразливе місце серверу або комп'ютеру, то користувач ризикує стати частиною мережі ботнету (машини які використовуються злочинцями у власних цілях без відома власника), втратою особистих даних або фінансових втрат. Одним з рішень передбачення подібних ситуацій виступає фаєрвол. Саме завдяки

вмілому налаштуванню мережевого екрану, є шанс знизити ризики до втрати цінних даних.

Також фаєрвол використовується для відкриття необхідних для роботи додаткам портів (наприклад 20-22 порти), та закриття невикористовуваних.

1.8. Висновки до першого розділу

При виборі способу розміщення веб-додатків або сайтів, користувачі мають широкий асортимент пропозицій від різноманітних сервісів, наприклад веб-хостинг, віртуальний сервер або виділений сервер. Вибір залежить від досвіду користувача в області адміністрування. Таким чином найчастіше вибір падає на віртуальний хостинг, так як більшість користувачів не мають достатнього досвіду у області адміністрування сервером, але користувачі, котрі планують розмістити серйозні комерційні проекти обирають виділені або віртуальні виділені сервери.

Найбільш збалансованим варіантом між ціною та функціоналом являються віртуальні сервери (VPS - virtual private server).

Головним критеріями обраного виду хостингу були:

- Повноцінний доступ з правами root;
- Можливість ручного встановлення різних версій ОС або ПО;
- Ціна оренди віртуальної машини;
- Виділена IP адреса.

Віртуальний веб-сервер створюється на базі фізичного виділеного серверу. VPS має необхідні можливості фізичного сервера, оперативну пам'ять і жорсткі диски або SSD, котрі розподілені від основного фізичного виділеного серверу. Кожен хостинг провайдер надає перелік можливих до

установки операційних систем. У нашому випадку були представлені Ubuntu, Debian, CentOS та Almalinux. Було обрано використання популярного стеку LAMP (Linux, Apache, MariaDB, PHP), для експлуатації нашого серверу у якості веб-хостингу і розміщення на ньому поширеної системи керування вмістом (Content Management System) -WordPress. Майбутня схема роботи серверу зображена на рисунку 1.5.

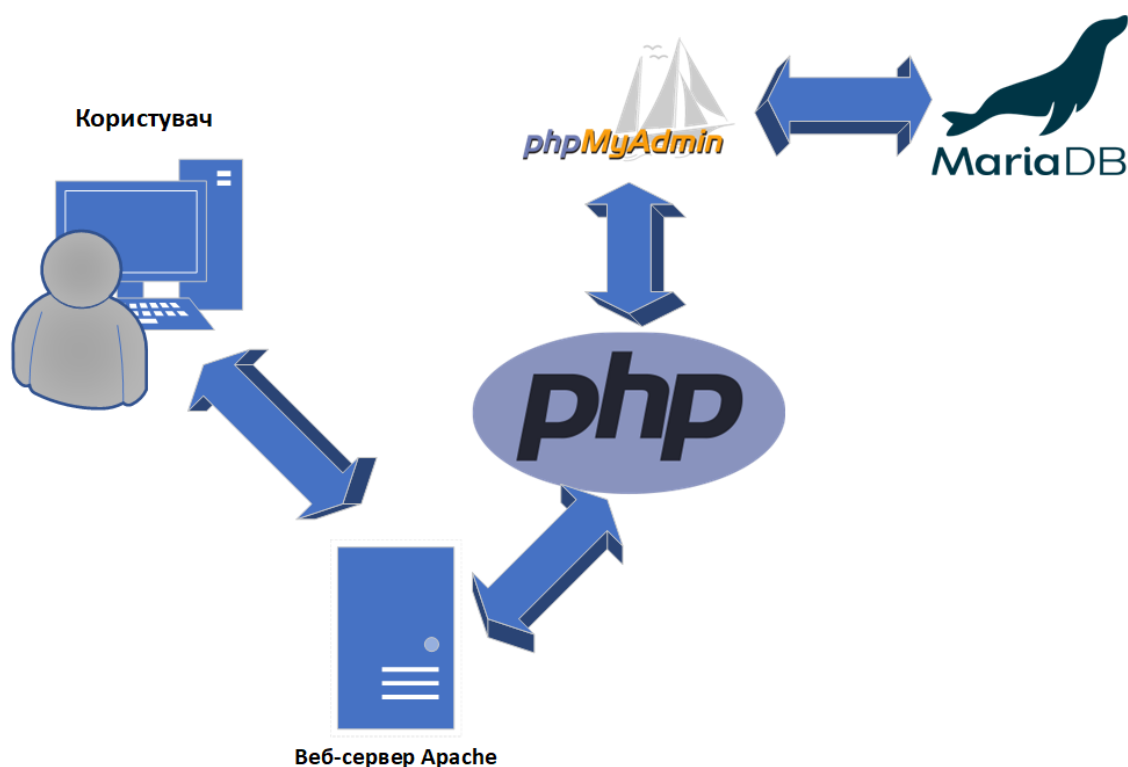


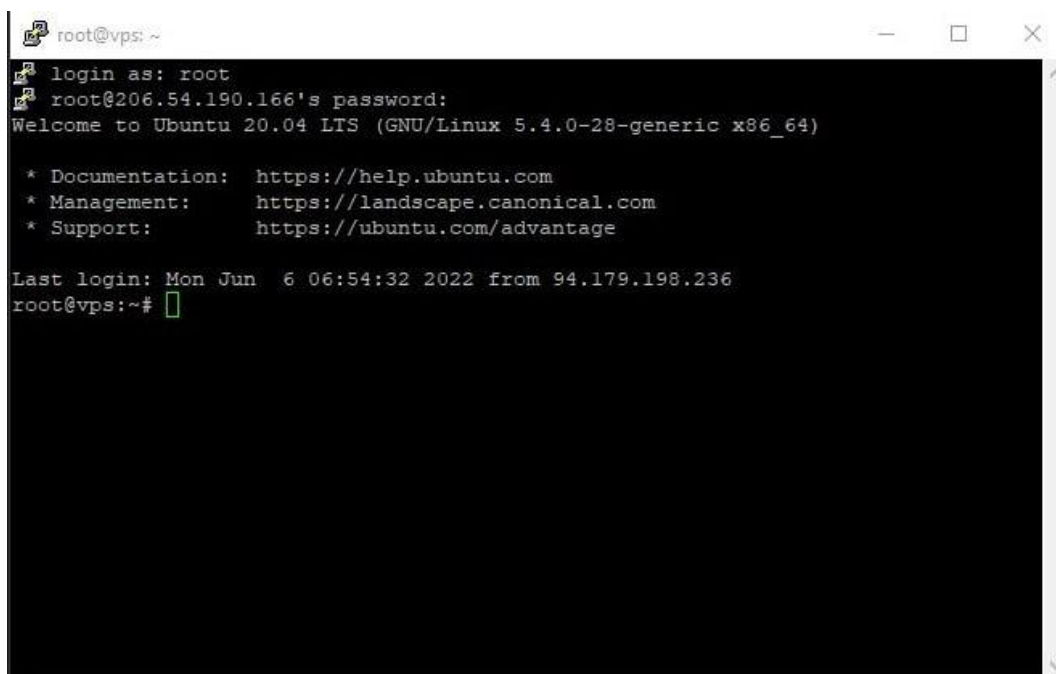
Рисунок 1.5 - Схема роботи LAMP-стеку на сервері

РОЗДІЛ 2

РОЗРОБКА ХМАРНОГО ХОСТІНГУ НА ОСНОВІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВІДКРИТИМ КОДОМ

2.1. Первинне налаштування виділеного серверу

За допомогою програмного забезпечення PuTTY підключились до нашого VPS серверу по протоколу SSH. Для з'єднання потрібні IP адреса серверу та root пароль. В нашому випадку підключення відбувається за стандартним 22 портом (Рисунок 2.1) .



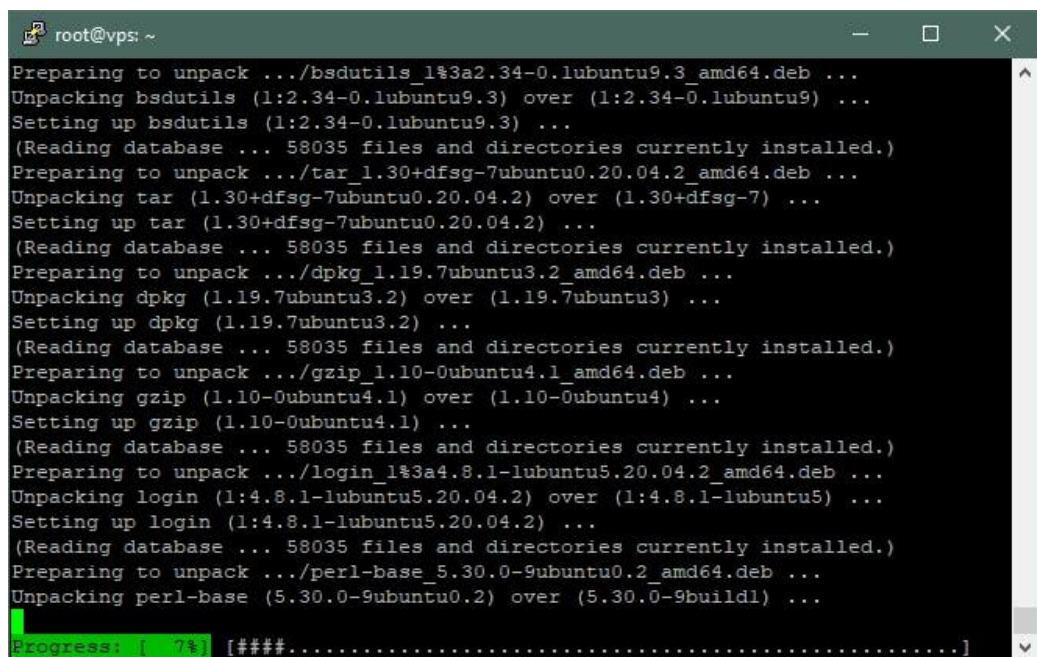
```
root@vps: ~  
login as: root  
root@206.54.190.166's password:  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-28-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
Last login: Mon Jun  6 06:54:32 2022 from 94.179.198.236  
root@vps:~#
```

Рисунок 2.1 - VPS сервер зі встановленою операційною системою Ubuntu 20.04

Після встановлення операційної системи або перед встановлення нового ПЗ, зазвичай проводиться оновлення індекс пакетів у системі Linux та оновлення пакетів програмного забезпечення до останніх версій. Виконували даний процес за допомогою команди

```
apt update && apt upgrade -y
```

Після оновлення використали команду `reboot` для перезавантаження системи (Рисунок 2.2).



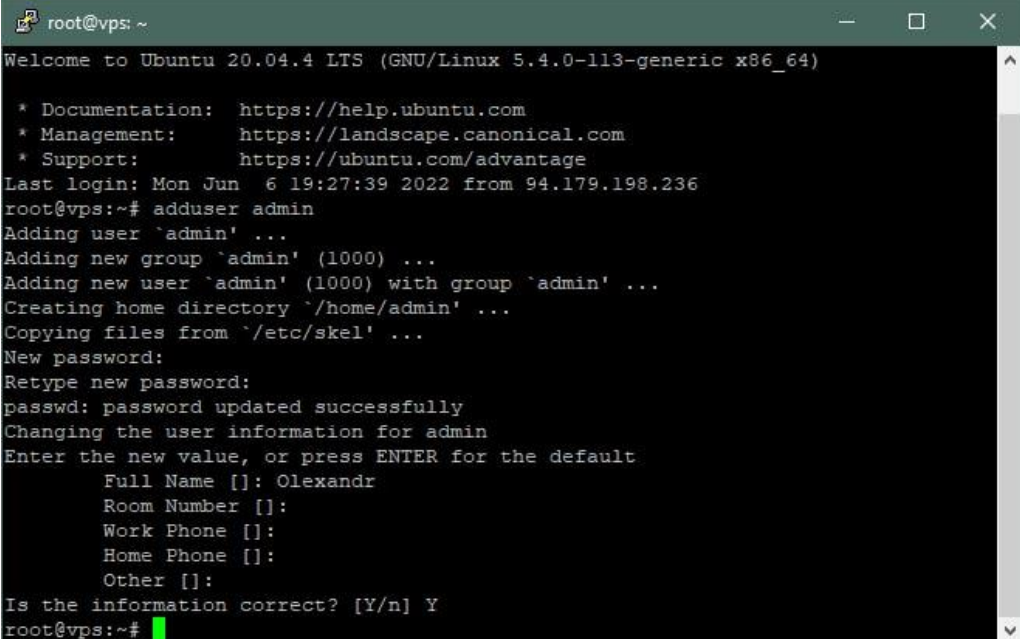
```
root@vps: ~
Preparing to unpack .../bsdutils_1%3a2.34-0.lubuntu9.3_amd64.deb ...
Unpacking bsutils (1:2.34-0.lubuntu9.3) over (1:2.34-0.lubuntu9) ...
Setting up bsutils (1:2.34-0.lubuntu9.3) ...
(Reading database ... 58035 files and directories currently installed.)
Preparing to unpack .../tar_1.30+dfsg-7ubuntu0.20.04.2_amd64.deb ...
Unpacking tar (1.30+dfsg-7ubuntu0.20.04.2) over (1.30+dfsg-7) ...
Setting up tar (1.30+dfsg-7ubuntu0.20.04.2) ...
(Reading database ... 58035 files and directories currently installed.)
Preparing to unpack .../dpkg_1.19.7ubuntu3.2_amd64.deb ...
Unpacking dpkg (1.19.7ubuntu3.2) over (1.19.7ubuntu3) ...
Setting up dpkg (1.19.7ubuntu3.2) ...
(Reading database ... 58035 files and directories currently installed.)
Preparing to unpack .../gzip_1.10-0ubuntu4.1_amd64.deb ...
Unpacking gzip (1.10-0ubuntu4.1) over (1.10-0ubuntu4) ...
Setting up gzip (1.10-0ubuntu4.1) ...
(Reading database ... 58035 files and directories currently installed.)
Preparing to unpack .../login_1%3a4.8.1-lubuntu5.20.04.2_amd64.deb ...
Unpacking login (1:4.8.1-lubuntu5.20.04.2) over (1:4.8.1-lubuntu5) ...
Setting up login (1:4.8.1-lubuntu5.20.04.2) ...
(Reading database ... 58035 files and directories currently installed.)
Preparing to unpack .../perl-base_5.30.0-9ubuntu0.2_amd64.deb ...
Unpacking perl-base (5.30.0-9ubuntu0.2) over (5.30.0-9build1) ...
Progress: [ 7%] [#####.....]
```

Рисунок 2.2 - Процес оновлення пакетів Linux

Команди `sudo` являється утилітою, яка надає права виконання адміністративних операцій, котрі доступні тільки для користувача `root`, для інших користувачів.

Створили користувача `admin` для безпеки управління системою (Рисунок 2.3), для якої можуть бути надлишковими або критичними, у випадку володіння привілегіями `root` користувача.

За допомогою команди `adduser admin` створили користувача і провели налаштування, щодо паролю та інформації про користувача.



```
root@vps: ~
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-113-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Mon Jun  6 19:27:39 2022 from 94.179.198.236
root@vps:~# adduser admin
Adding user `admin' ...
Adding new group `admin' (1000) ...
Adding new user `admin' (1000) with group `admin' ...
Creating home directory `/home/admin' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for admin
Enter the new value, or press ENTER for the default
   Full Name []: Olexandr
   Room Number []:
   Work Phone []:
   Home Phone []:
   Other []:
Is the information correct? [Y/n] Y
root@vps:~#
```

Рисунок 2.3 - Процес створення користувача `admin`

Наділили користувача `admin` правами суперкористувача, додаючи його у групу `sudo` (Рисунок 2.4) командою

```
usermod -aG sudo admin
```

```
root@vps:~# usermod -aG sudo admin
root@vps:~# █
```

Рисунок 2.4 - Успішне виконання команди

Ключ `-a`, для команди `usermod`, додає користувача в додаткову групу і використовується з ключем `G`, котрий перераховує список груп, у яких буде додано користувача.

Перевірили, чи додався користувач до групи. Для цього перемикнулись з користувача `root` на користувача `admin` (Рисунок 2.1.5). Виконували за допомогою команди

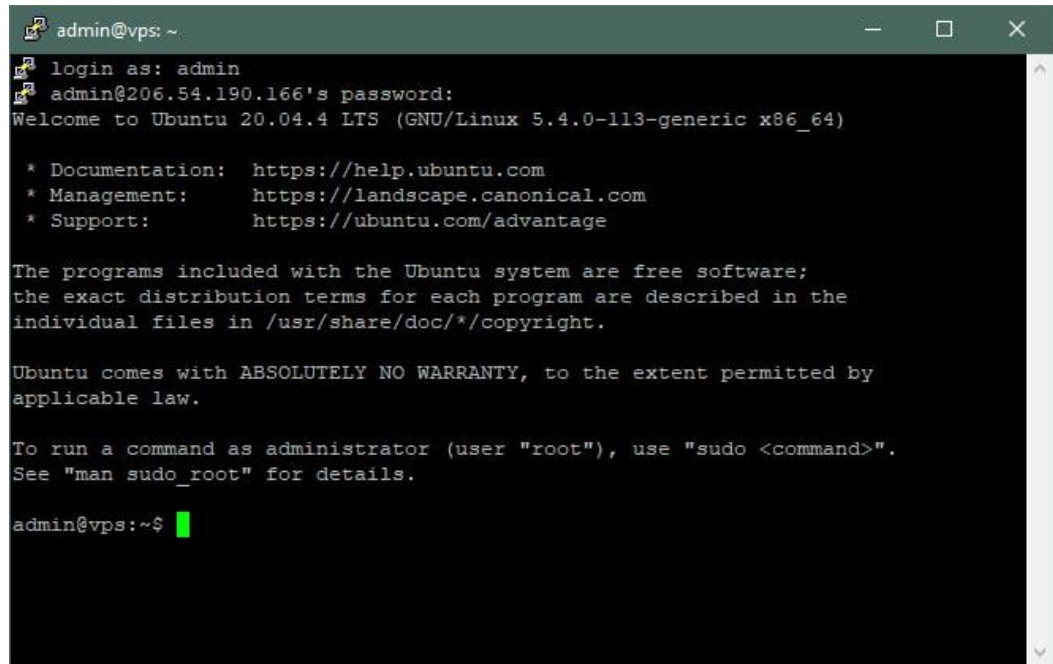
```
su admin
```

```
root@vps:~# su admin
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@vps:/root$ █
```

Рисунок 2.5 - Перевірили наявність користувача `admin`

Повторно приєднались до серверу по протоколу `SSH` з доступами створеного користувача `admin` (Рисунок - 2.6).



```
admin@vps: ~  
login as: admin  
admin@206.54.190.166's password:  
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-113-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
admin@vps:~$
```

Рисунок 2.6 - Успішне підключення користувачем admin

2.2. Встановлення та конфігурування веб-серверу Apache

Почали інсталювання стека LAMP з веб-серверу Apache за допомогою наступної команди :

```
sudo apt-get install apache2
```

Перейшли у веб-браузері на адресу IP серверу для перевірки успішного інсталювання Apache (Рисунок 2.7).

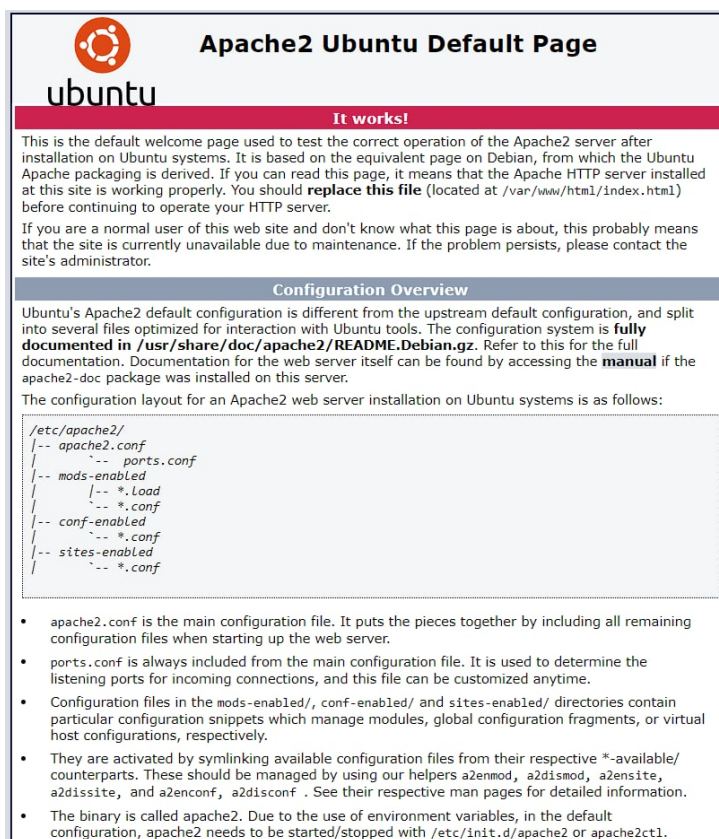


Рисунок 2.7 - Стандартна сторінка Apache, яка відображається при переході на IP адресу серверу

Створили директорію `public_html`, для розміщення в ній майбутнього контенту сайту, за допомогою базової команди `mkdir`. Далі надали привілежії для створеної директорії `755` та права `admin`, щоб даний користувач міг редагувати вміст директорії без утиліти `sudo` (Рисунок - 2.2.2).

```

admin@vps:~$ sudo mkdir /var/www/public_html
admin@vps:~$ ls /var/www
html public_html
admin@vps:~$ sudo chmod -R 744 /var/www
-bash: sudo: command not found
admin@vps:~$ sudo chmod -R 744 /var/www
admin@vps:~$ sudo chmod -R 755 /var/www
admin@vps:~$ sudo chown -R admin:admin /var/www
admin@vps:~$ █

```

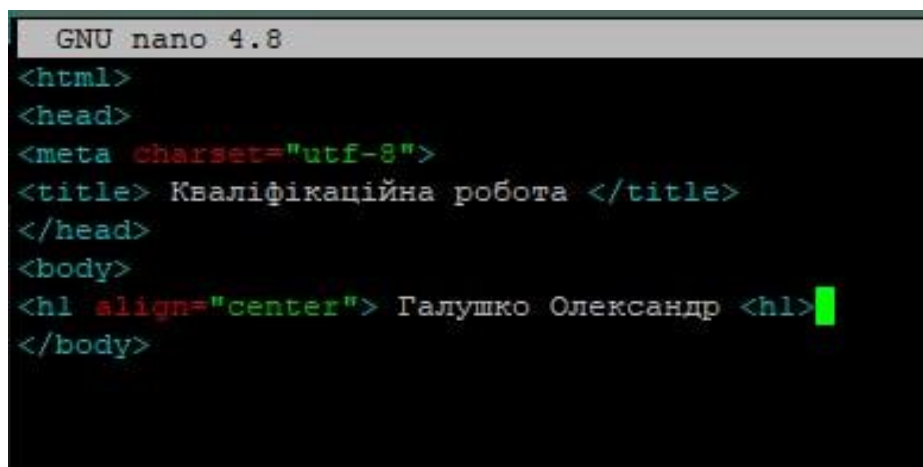
Рисунок 2.8 – Процес створення директорії та надання привілежій для користувача `admin`

Створили простий .html файл (Рисунок 2.9) для перевірки роботи нашого веб-серверу, за допомогою попередньо інстальованого текстового редактору nano.

Команда має наступний вигляд :

```
nano /var/www/public_html/index.html
```

Дана команда при відсутності файлу, автоматично створює його та відкриває редактор.



```
GNU nano 4.8
<html>
<head>
<meta charset="utf-8">
<title> Кваліфікаційна робота </title>
</head>
<body>
<h1 align="center"> Галушко Олександр </h1>
</body>
```

Рисунок 2.9 – Тестовий файл index.html

Для зручного пошуку або виконання дій з файлами використовуємо, попередньо встановлений, візуальний файловий менеджер mc – Midnight Commander (Рисунок 2.10).

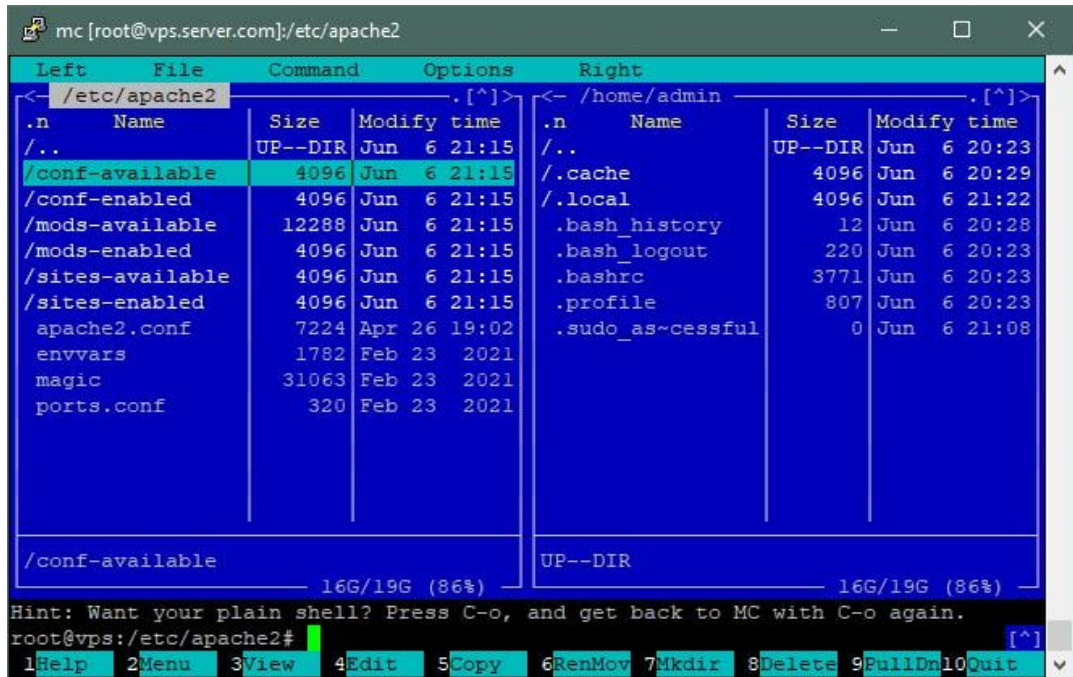


Рисунок 2.10 – Інтерфейс файлового менеджера MC

При переході на IP адресу нашого серверу відкривається стандартна сторінка Apache. Це значить що необхідно створити конфігураційний файл для Apache та створити віртуальний хост .

Так як у нас відсутнє зареєстроване доменне ім'я, ми можемо додати наш «домен» до файлу hosts на локальному комп'ютері і продовжувати роботу над нашим сайтом. Так наш комп'ютер буде знати що IP адреса нашого сервера відповідає нашому «домону» `diplom1`. Для цього додали запис «*IP адреса сервер* `diplom1`» до файлу hosts на локальному комп'ютері. Значення `diplom1` буде виступати в ролі нашого домену.

Для створення конфігураційного файлу Apache, ми скопіювали стандартний файл зі збереженням привілеїв з відповідної директорії та перейменували його (Рисунок 2.11), для подальшого редагування відповідно наших цілей наступною командою:

```
sudo cp -p /etc/apache2/sites-available/000-
defaulttt.conf /etc/apache2/sites-available/
public_html.conf
```

```
admin@vps:~$ ls /etc/apache2
apache2.conf  conf-enabled  envvars  magic  mods-available  mods-enabled  ports.conf  sites-available  sites-enabled
admin@vps:~$ ls /etc/apache2/sites-available
000-default.conf  default-ssl.conf
admin@vps:~$ sudo cp -p /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/public_html.conf
```

Рисунок 2.11 – Процес створення конфігураційного файлу Apache

Далі було відредаговано новостворений файл, де вказали, в якій директорії буде розташовано сайт `diplom1` (Рисунок 2.12).

```
GNU nano 4.8
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName diplom1

ServerAdmin webmaster@localhost
DocumentRoot /var/www/public_html

# Available loglevels: trace8, ..., tracel, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

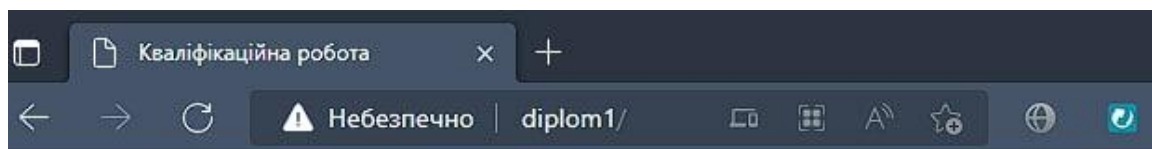
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Рисунок 2.12 – Відредагований конфіг файл Apache.

Після зроблених процедур перезавантажили Apache для збереження налаштувань командою

```
sudo service apache2 restart.
```

Після цього перевірили доступність нашого сайту за адресою <http://diplom1> (Рисунок 2.13)



Галушко Олександр

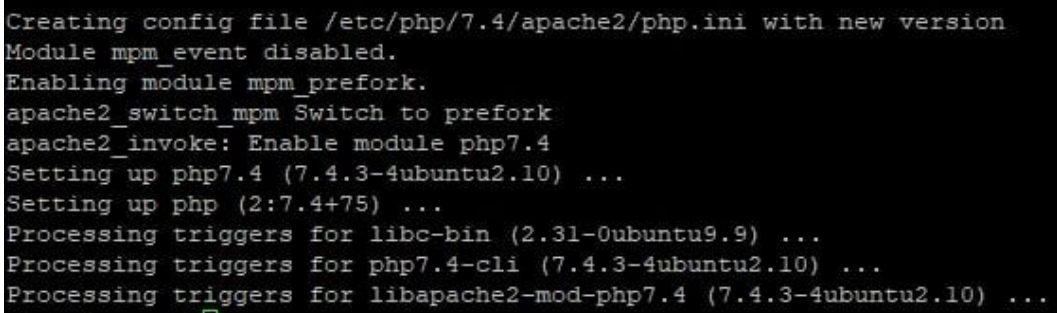
Рисунок 2.13 – Відображення створеного файлу html

Після успішного інсталювання та налаштування Apache перейшли до встановлення пакету PHP.

2.3. Інсталювання PHP

Інсталювали PHP за допомогою команди (Рисунок 2.14):

```
sudo apt-get install php
```



```
Creating config file /etc/php/7.4/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
apache2_invoke: Enable module php7.4
Setting up php7.4 (7.4.3-4ubuntu2.10) ...
Setting up php (2:7.4+75) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.10) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.10) ...
```

Рисунок 2.14 – Процес інсталювання PHP

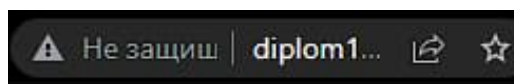
Перевіряємо роботу PHP за допомогою файлу test.php (Рисунок 2.15), який було розташовано у кореневій директорії сайту public_html.

Код файлу test.php мав наступний вигляд:

```
<?php
```

```
    echo "<center> PHP Work</center>";
```

```
?>
```





PHP Work

Рисунок 2.15 – Відображення файлу test.php по адресі <http://diplom1/test.php>

Для визначення інформації про версію PHP (Рисунок 2.16) та завантажених модулів було змінено код файлу test.php на наступний:

```
<?php
phpinfo(INFO_MODULES);
?>
```

PHP Version 7.4.3 	
System	Linux vps.server.com 5.4.0-113-generic #127-Ubuntu SMP Wed May 18 14:30:56 UTC 2022 x86_64
Build Date	Mar 2 2022 15:36:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysmsg.ini, /etc/php/7.4/apache2/conf.d/20-system.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokentizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*
<small>This program makes use of the Zend Scripting Language Engine: Zend Engine v3.4.0, Copyright (c) Zend Technologies with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies</small> 	

Configuration

apache2handler

Рисунок 2.16 – Відображення відредагованого файлу test.php по адресі

<http://diplom1/test.php>

Після успішної перевірки роботи PHP перейшли до встановлення наступного компоненту нашого серверу - MariaDB.

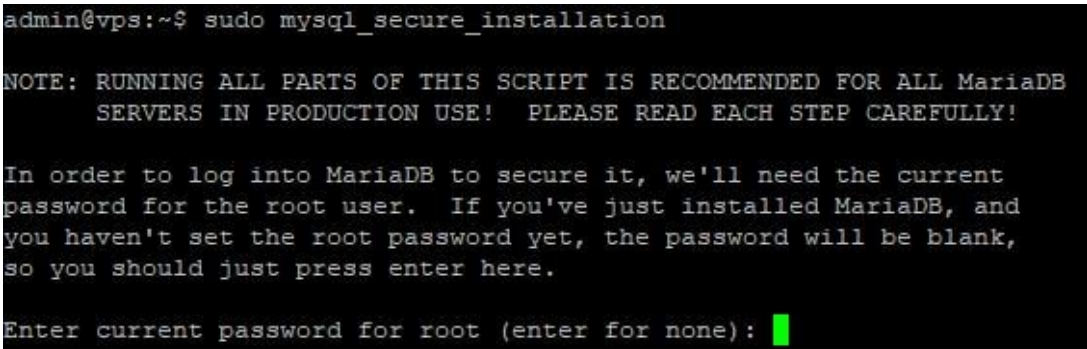
2.4 Інсталювання і конфігурація СУБД MariaDB

Інсталювали сервер та клієнт MySQL MariaDB за допомогою команди:

```
sudo apt-get install mariadb-server mariadb-client
```

та запустили налаштування mysql-серверу (Рисунок 2.17) командою

```
sudo mysql_secure_installation
```

A terminal window showing the execution of the 'mysql_secure_installation' command. The prompt is 'admin@vps:~\$'. The output includes a note about production use, instructions on setting a root password, and a prompt for the current password for root. A green cursor is visible at the end of the last line.

```
admin@vps:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none): █
```

Рисунок 2.17 – Початок налаштування серверу mysql

Після встановлення паролю та повного завершення налаштувань підключились та перевірили наявність бази даних (Рисунок 2.18) наступними командами:

```
mysql -uroot -p
```

```
show database
```

```
admin@vps:~$ mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 46
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema     |
+-----+
3 rows in set (0.001 sec)

MariaDB [(none)]> █
```

Рисунок 2.18 – демонстрація робочої бази даних MySQL

Для більш зручного управління базами даних було проведено інсталювання популярного веб-додатку з відкритим кодом та графічним інтерфейсом - phpMyAdmin.

Інсталювання проводилось за допомогою команди

```
sudo apt-get install phpmyadmin php-mbstring php-gettext
```

Після проведення початкового налаштування для Apache, перевірили доступність phpMyAdmin за адресою `http://diplom1/phpMyAdmin/` (Рисунок 2.19).

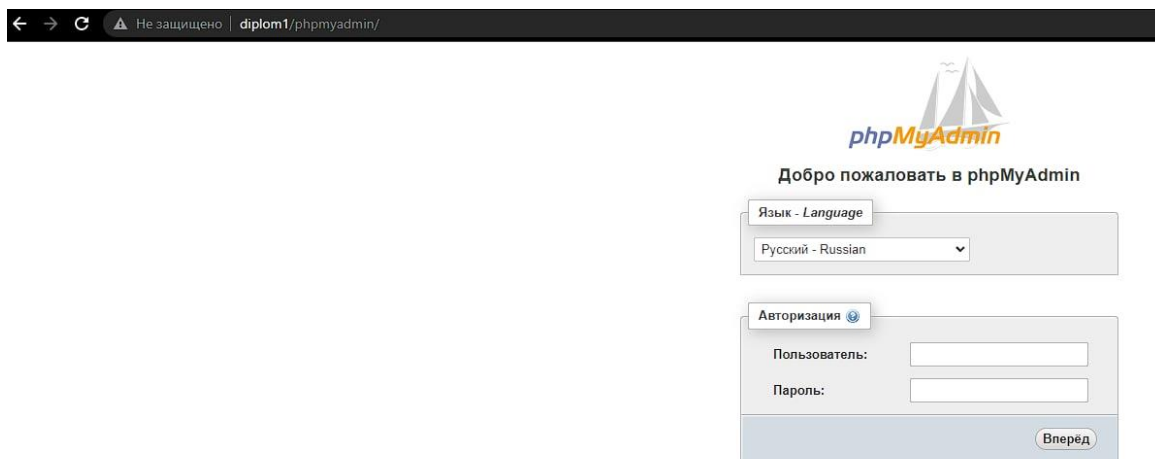


Рисунок 2.19 – Демонстрація доступного веб-додатку phpMyAdmin на нашому сервері

Використавши дані при інсталюванні зайшли до робочого інтерфейсу phpMyAdmin (Рисунок 2.20).

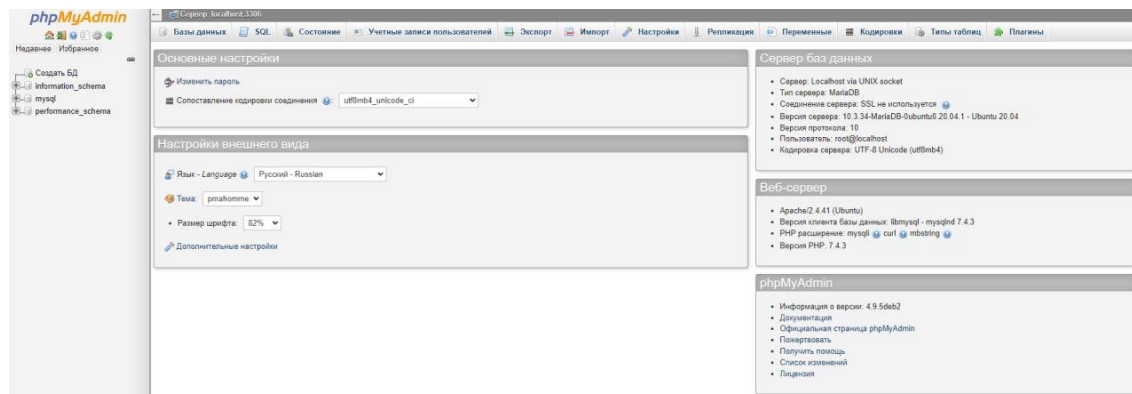


Рисунок 2.20 – Інтерфейс веб-додатку phpMyAdmin

Для перевірки коректного налаштування веб-серверу встановили систему керування веб сайтами з відкритим кодом WordPress.

2.5. Встановлення системи керування вмістом WordPress

Було вирішено провести скачування архіву WordPress прямо на сервер за допомогою команди

```
wget https://wordpress.org/latest.tar.gz
```

Перевірили наявність архіву в домашній директорії та розпакували вміст архіву (Рисунок 2.21) за допомогою команди :

```
tar -xzf latest.tar.gz
```



```
admin@vps:~$ ls
latest.tar.gz
admin@vps:~$ tar -xzf latest.tar.gz
admin@vps:~$ ls
latest.tar.gz  wordpress
admin@vps:~$
```

Рисунок 2.21 – Демонстрація наявності архіву та його розпакування

За допомогою файлового менеджера mc перейшли до кореневої (Рисунок 2.22) папки сайту та видалили файли минулого сайту .

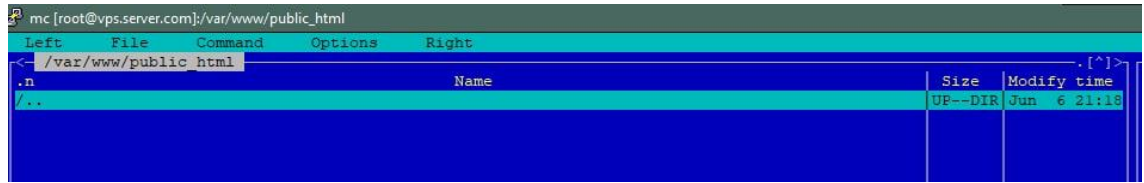


Рисунок 2.22 - Наявність очищеної кореневої директорії сайту

Перемістили вміст папки wordpress з домашньої директорії до кореневої папки (Рисунок 2.23) за допомогою команди

```
mv wordpress/* /var/www/public_html
```

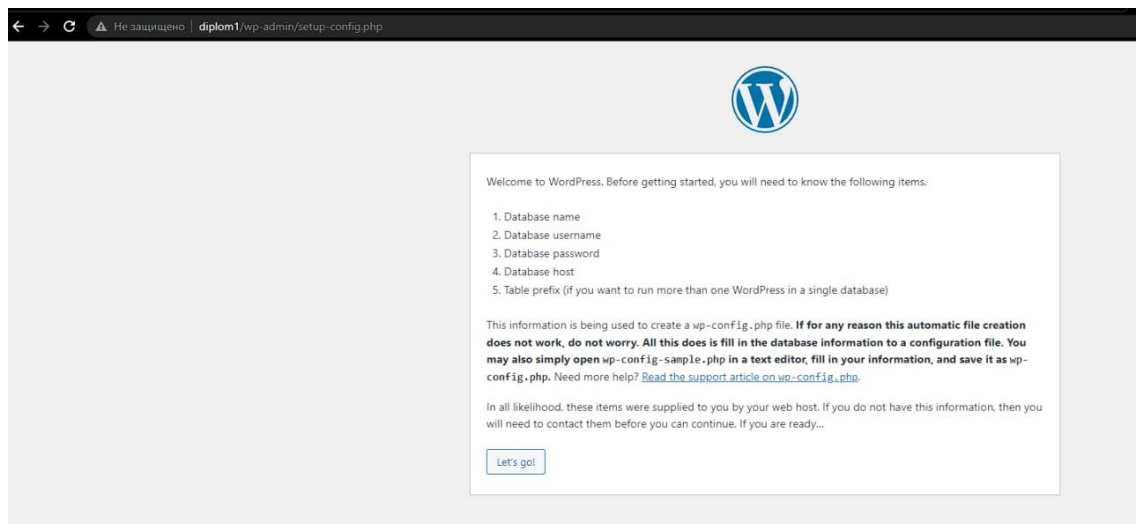


Рисунок 2.23 – Демонстрація вступної сторінки системи WordPress при переході на сайт <http://diplom1>

Для початку налаштування WordPress, було створено відповідну базу даних. Створили базу даних за допомогою інтерфейсу phpMyAdmin (Рисунок 2.5.4).

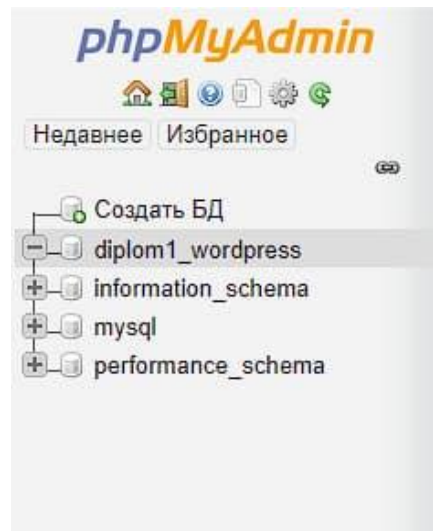


Рисунок 2.24 – Демонстрація створеної бази даних для WordPress

Для того щоб можливо було користуватись базою даних, було створено користувача admin, якому були надані повні привілегиї у веб-додатку phpMyAdmin (Рисунок 2.25).

Також при наявності дампу бази даних за допомогою цього веб-додатку можливо здійснити імпорт, або ж навпаки експортувати наявну базу даних на комп'ютер або на сервер. Доступний і моніторинг бази даних у відповідних вкладках веб-додатку.

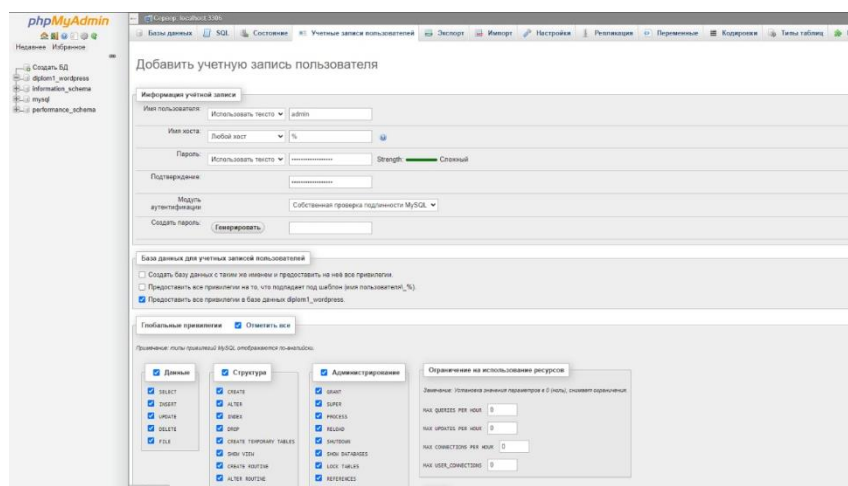


Рисунок 2.25 – Інтерфейс створення користувача у веб-додатку phpMyAdmin

Було продовжено інсталювання WordPress з використанням вже створеної бази даних, користувача та його паролю. Поля для вказання хосту і префіксу таблиць залишили без змін (Рисунок 2.26).

Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name: The name of the database you want to use with WordPress.

Username: Your database username.

Password: Your database password.

Database Host: You should be able to get this info from your web host. If localhost does not work.

Table Prefix: If you want to run multiple WordPress installations in a single database, change this.

Рисунок 2.26 – Інсталювання Wordpress

Наступними кроками було вказання у конфіг файлі Wordpress дані бази даних та користувача.

Створили файл конфігу (Рисунок 2.27) у кореневій директорії сайту та додаємо код, котрий було представлено нам при інсталюванні WordPress командою

```
nano /var/www/public_html/wp-config.php
```

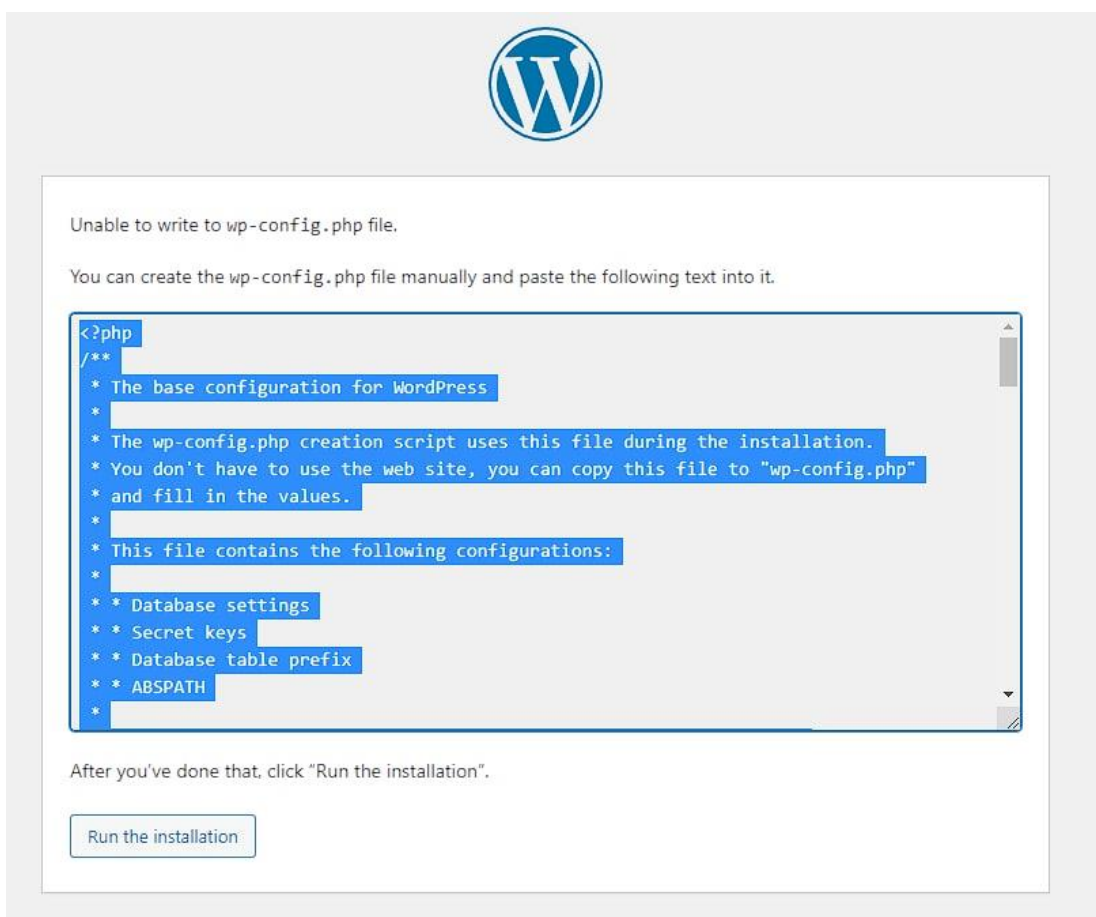


Рисунок 2.27 – Вміст конфіг файлу Wordpress

Продовжили інсталювання WordPress, створивши користувача для самої системи WordPress (Рисунок 2.28).

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password
Very weak
Important: You will need this password to log in. Please store it in a secure location.

Confirm Password Confirm use of weak password

Your Email
Double-check your email address before continuing.

Search engine visibility Discourage search engines from indexing this site
It is up to search engines to honor this request.

Рисунок 2.28 – Інтерфейс створення користувача у WordPress

Після завершення інсталювання Wordpress відредагували головну сторінку сайту та перевірили коректність роботи сайту по адресі <http://diplom1> (Рисунок 2.29).

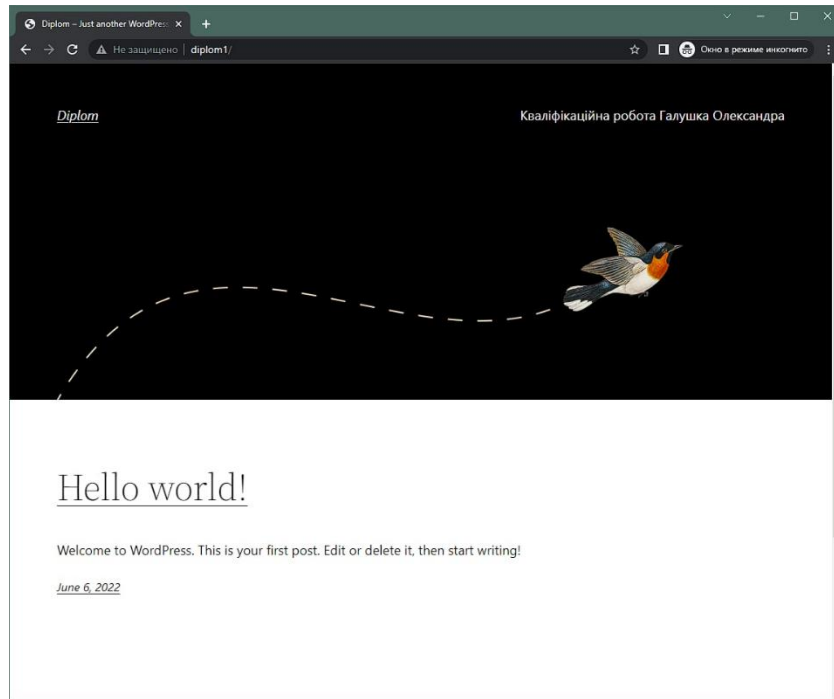


Рисунок 2.29 – Демонстрація роботи сайту на налаштованому нами сервері

2.6. Налаштування протоколу ftp

Для зручної загрузки файлів сайту для користувача було встановлено (Рисунок 2.30) та налаштовано ftp-server за допомогою команди

```
sudo apt install vsftpd
```

```

admin@vps:~$ sudo apt install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  linux-image-4.15.0-99-generic linux-modules-4.15.0-99-generic
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 115 kB of archives.
After this operation, 338 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/main amd64 vsftpd amd64 3.0.3-12 [115 kB]
Fetched 115 kB in 1s (184 kB/s)
Preconfiguring packages ...
Selecting previously unselected package vsftpd.
(Reading database ... 103253 files and directories currently installed.)
Preparing to unpack .../vsftpd_3.0.3-12_amd64.deb ...
Unpacking vsftpd (3.0.3-12) ...
Setting up vsftpd (3.0.3-12) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service - /lib/systemd/system/vsftpd.service.
Processing triggers for systemd (245.4-4ubuntu3.17) ...
admin@vps:~$ █

```

Рисунок 2.31 – Процес встановлення vsftpd

Провели налаштування vsftpd у конфіг файлі (Рисунок 2.32) командою:

```
sudo nano /etc/vsftpd.conf
```

В ньому ми організували дозвіл на доступ для вказаних у файлі vsftpd.chroot_list, прослуховування вхідних підключень, можливість запису файлів на сервері та інші.


```

admin@vps:~$ sudo ss -ltnp
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
Process
LISTEN    0           80          127.0.0.1:3306           0.0.0.0:*
  users: (("mysqld",pid=12111,fd=21))
LISTEN    0          4096        127.0.0.53%lo:53        0.0.0.0:*
  users: (("systemd-resolve",pid=258,fd=13))
LISTEN    0           128         0.0.0.0:22              0.0.0.0:*
  users: (("sshd",pid=415,fd=3))
LISTEN    0           511         *:80                    *:80
  users: (("apache2",pid=45397,fd=4), ("apache2",pid=41080,fd=4), ("apache2",pid=40772,fd=4), ("apache2",pid=40771,fd=4), ("apache2",pid=40770,fd=4), ("apache2",pid=40769,fd=4), ("apache2",pid=40768,fd=4), ("apache2",pid=11097,fd=4))
LISTEN    0           128         [::]:22                 [::]:*
  users: (("sshd",pid=415,fd=4))
admin@vps:~$

```

Рисунок 2.33 – Стан відкритих портів серверу

Після того як ми впевнились що потрібні порти закриті, інстальювали інтерфейс управління брандмауером `ufw`, за допомогою команди

```
sudo apt install ufw
```

Перед включенням брандмауера переконались, що підключення OpenSSH дозволене (Рисунок 2.34), командою

```
sudo ufw status
```

```

admin@vps:~$ sudo ufw status
Status: active

To Action From
--
OpenSSH ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)

```

Рисунок 2.34 - Статус `ufw`

Після цього відкрили необхідні порти (Рисунок 2.35). В нашому випадку 20-21 для ftp, 22 для ssh та 80 для Apache командами

```
sudo ufw allow *номер порту*
```

```
admin@vps:~$ sudo ufw status
Status: active

To Action From
--
OpenSSH ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)

admin@vps:~$ sudo ufw allow 21
Rule added
Rule added (v6)
admin@vps:~$ sudo ufw allow 22
Rule added
Rule added (v6)
admin@vps:~$ sudo ufw allow 80
Rule added
Rule added (v6)
admin@vps:~$ sudo ufw status
Status: active

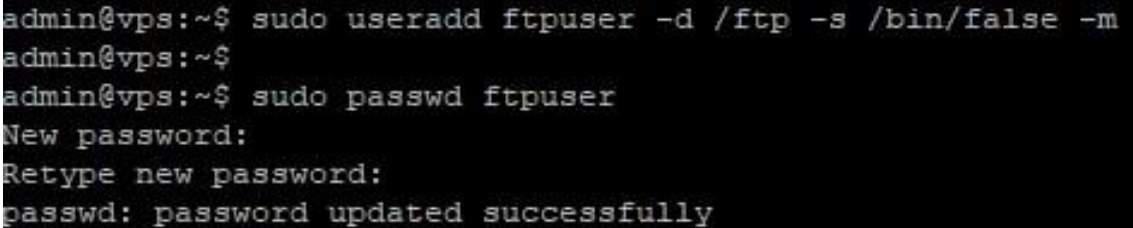
To Action From
--
OpenSSH ALLOW Anywhere
21 ALLOW Anywhere
22 ALLOW Anywhere
80 ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)
21 (v6) ALLOW Anywhere (v6)
22 (v6) ALLOW Anywhere (v6)
80 (v6) ALLOW Anywhere (v6)

admin@vps:~$ █
```

Рисунок 2.35 - Оновлений стан відкритих портів серверу

Створили окремого ftp-користувача ftpuser командою та задали йому пароль (Рисунок 2.36) командами:


```
useradd ftpuser -d /ftp -s /bin/false -m  
  
sudo passwd ftpuser
```



```
admin@vps:~$ sudo useradd ftpuser -d /ftp -s /bin/false -m  
admin@vps:~$  
admin@vps:~$ sudo passwd ftpuser  
New password:  
Retype new password:  
passwd: password updated successfully
```

Рисунок 2.36 – Успішне створення нового користувача

Задали відповідні права користувачеві, раніше використовуваними командами, та виконали перезавантаження vsftpd командою

```
sudo systemctl restart vsftpd
```

Після цього підключились за допомогою програми FileZilla з власного комп'ютера з ОС Windows (Рисунок 2.37).

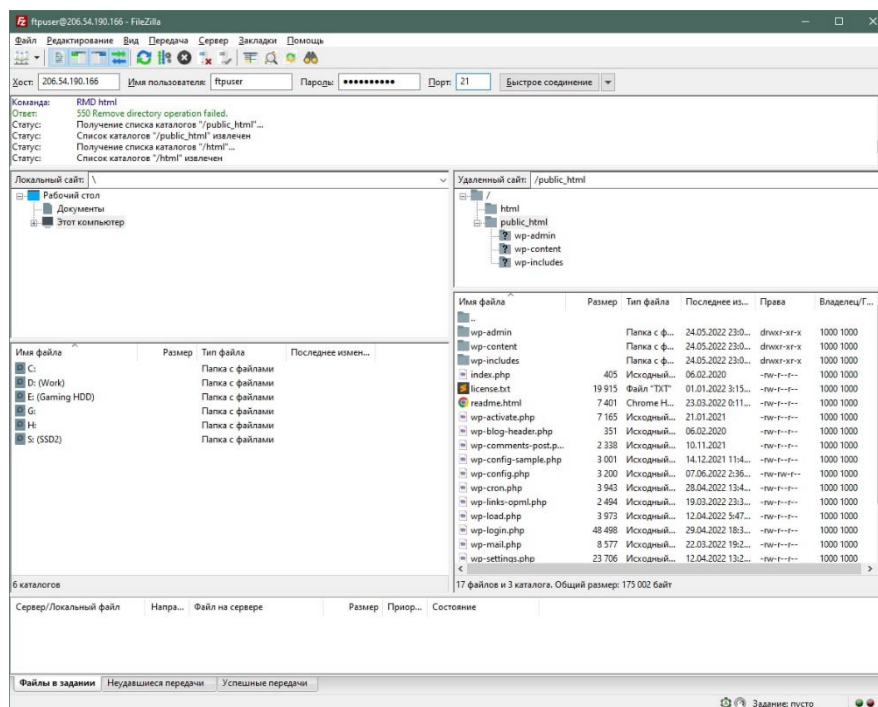


Рисунок 2.37 - Демонстрація успішного підключення по протоколу ftp

Перевірили можливість підключення до ftp-серверу, підключившись за нашої VPS (Рисунок 2.38).

```
admin@vps:~$ ftp localhost
Connected to localhost.
220 (vsFTPd 3.0.3)
Name (localhost:admin): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x  2 1001  1001    4096 Jun 09 22:00 html
drwxr-xr-x  5 1001  1001    4096 Jun 06 23:36 public_html
```

Рисунок 2.38 - Демонстрація успішного підключення з серверу

Для перевірки наявності прав у користувача ftpuser спробували передати файл з комп'ютеру на сервер за допомогою ftp через програму FileZilla.

Створили текстовий файл Галушко О.Г.126-18-1.txt та завантажили його на сервер використовуючи інтерфейс програми (Рисунок 2.39).

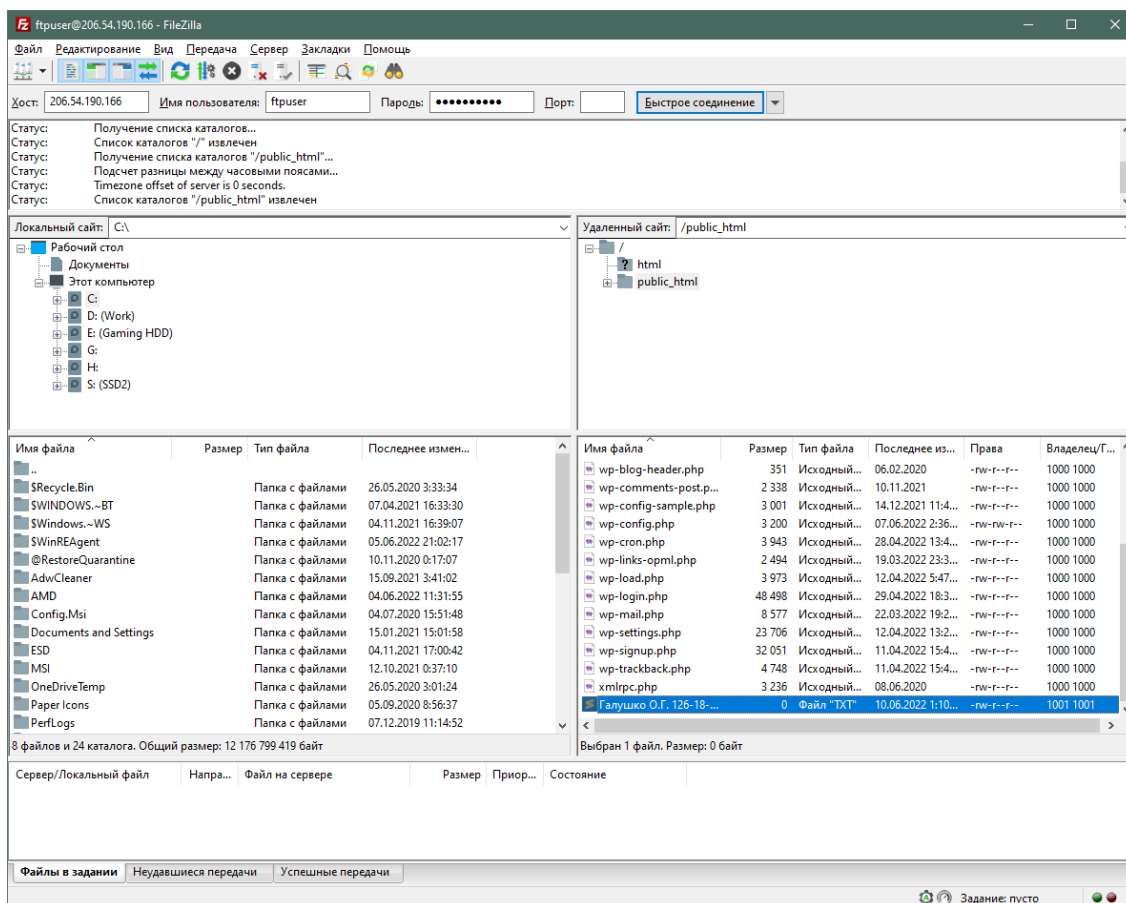


Рис 2.39 - Демонстрація успішного копіювання файлу на сервер.

Також спробували підключитись за допомогою більш безпечного протоколу sftp, який виконує операції поверх надійного і безпечного з'єднання з шифруванням (Рисунок 2.40).

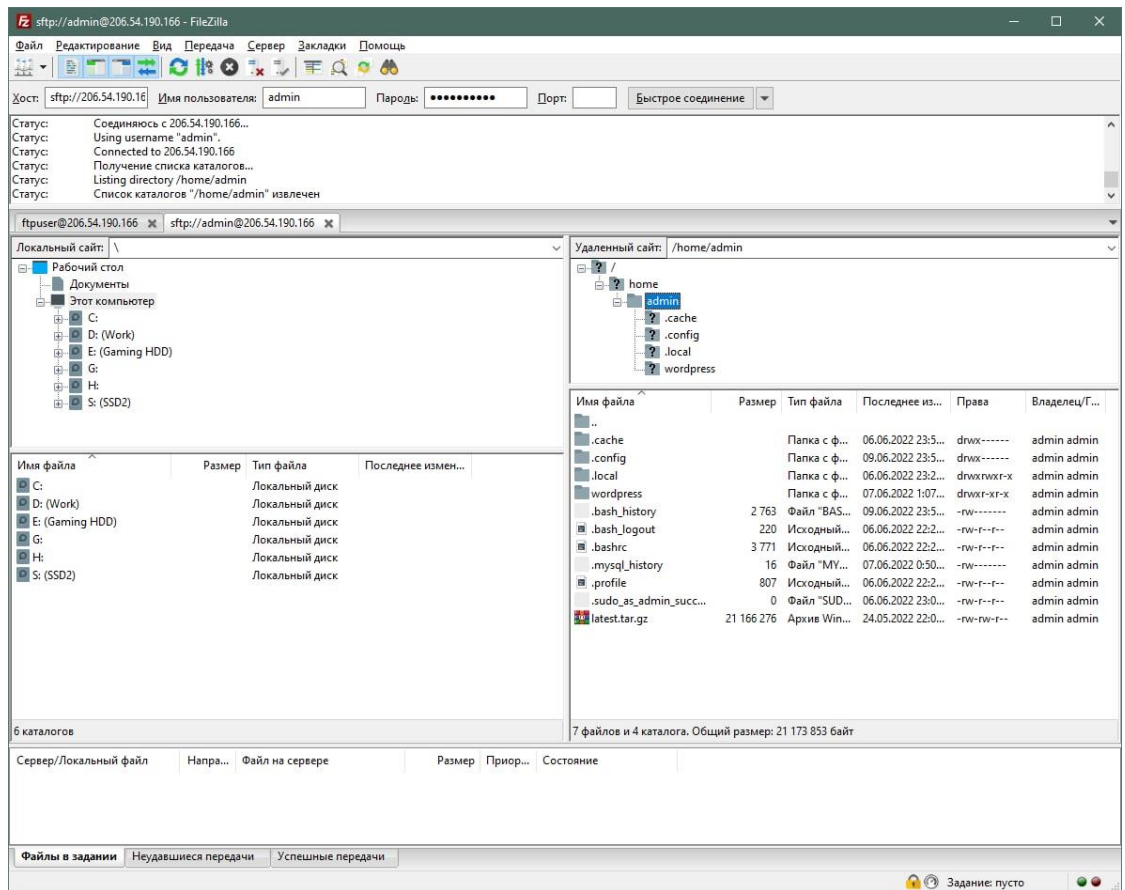


Рисунок 2.40 - Демонстрація підключення по sftp

2.7. Висновки до другого розділу

Таким чином отримали повністю налаштований веб-хостинг для використання його в мережі Інтернет. Сервер має власну виділену IP адресу, налаштовані протоколи для підключення, сконфігурований фаєрвол для підвищення безпеки, зручний інтерфейс для управління базами даних phpMyAdmin та інші інструменти для комфортного використання хостингом.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було продемонстровано аналіз стану області рішень, відповідно до якого, було проведено огляд видів хостингів існуючих на сьогодні. Крім того, було виконано аналіз існуючих рішень та програмного забезпечення для організації власного хмарного хостингу, який допоміг чітко поставити необхідні завдання для організації роботи системи в мережі Інтернет.

Задачі, які було поставлено на початку роботи, сформулювали подальший напрямок у налаштуванні та організації виділеного віртуального серверу та його додатків для використання у якості хмарного хостингу.

По завершенню аналізу варіантів оптимального набору програмного забезпечення, конфігурацій та додатків для серверу, було проведено налаштування серверу відповідно до поставлених задач.

В результаті було отримано повністю налаштований сервер для підключення домену та використання сайту у мережі. Користувач хостингу отримує повністю налаштований веб-сервер з можливістю вдосконалення налаштувань у майбутньому, завантаження свого сайту або розробки нового за допомогою CMS Wordpress, імпорту бази даних через підключення по протоколу SSH або за допомогою інтерфейсу веб-додатку phpMyAdmin, управління фаєрволом та інші можливості root користувача в на сервері.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Важливість власного веб-сайту сьогодні [Електронний ресурс] – Режим доступу до ресурсу: <https://boshnikoff.com/zachem-nuzhen-sajt/#i>
2. Визначення хмарного хостингу [Електронний ресурс] – Режим доступу до ресурсу: <https://tebapit.com/%D1%89%D0%BE-%D0%BD%D0%B0%D1%81%D0%BF%D1%80%D0%B0%D0%B2%D0%B4%D1%96-%D1%94-%D1%85%D0%BC%D0%B0%D1%80%D0%BD%D0%B8%D0%B9-%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3/>
3. Визначення хостингу [Електронний ресурс] – Режим доступу до ресурсу: <http://www.tops.org.ua/uk/shcho-take-veb-khostynh.html>
4. Віртуальний виділений сервер [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%92%D1%96%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D0%B8%D0%B9_%D0%B2%D0%B8%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80
5. Фізичний виділений сервер [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%92%D0%B8%D0%B4%D1%96%D0%BB%D0%B5%D0%BD%D0%B8%D0%B9_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80
6. Веб сервер [Електронний ресурс] – Режим доступу до ресурсу: <https://interweber.ru/technology/dlya-chego-sluzhit-veb-server-programmy-i-programmnoe-obespechenie.html>
7. Аналіз використання веб-серверів [Електронний ресурс] – Режим доступу до ресурсу: <https://www.similartech.com/compare/apache-vs-nginx>

8. Порівняння СУБД [Електронний ресурс] – Режим доступу до ресурсу: <https://webhosting.de/ru/mariadb-the-interesting-database-alternative-to-mysql/>

9. Визначення PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/PHP>

10. Характеристика протоколу SSH [Електронний ресурс] – Режим доступу до ресурсу: <https://guides.hexlet.io/ru/ssh/>

11. Характеристика протоколу s/ftp [Електронний ресурс] – Режим доступу до ресурсу: https://www.ukraine.com.ua/blog/hosting_ukraine/chto-takoe-ftp-client-i-kak-im-polzovatsya.html

12. Принцип роботи фаєрволу [Електронний ресурс] – Режим доступу до ресурсу: <https://selectel.ru/blog/tutorials/how-to-configure-firewall-with-ufw-on-ubuntu-20/>

13. SSH, The Secure Shell: The Definitive Guide / Daniel J. Barrett, Richard E. Silverman, Robert G. Byrnes, 2005. - 668 с.

ДОДАТОК А

Відомість матеріалів кваліфікаційної роботи

№ з/п	Позначення				Найменування	Кільк. аркушів	Примітки		
1									
2					Документація				
3									
4	ІТКІ.КР.22.02.ДА.ПЗ				Пояснювальна записка		Формат А4		
5									
6					Презентація				
7									
8					Диск CD-R з презентацією	1	Диск CD-R		
9									
					ІТКІ.КР.22.02.ДА.ПЗ.				
Зм.	Ар-куш	№ докум.	Підпис	Дата					
Розроб.	Галушко О.Г.				Матеріали кваліфікаційної роботи	Літ.		Аркуш	Аркушів
Керівник	Соколова Н.О.					Н		1	1
Рецензент	<i>Гуліна І.Г.</i>					НТУ «ДП», 12; 126-18-1			
Н.контр.	Коротенко Г.М.								
Зав. каф.	Гнатушенко В.В.								