Лабораторна робота №21

Розгортання та базове налаштування програмного стеку LAMP для вебзастосунків на базі Ubuntu Server

Мета: розгорнути та виконати базове налаштування програмного стеку LAMP на базі Ubuntu Server, забезпечивши функціонування вебсервера Apache, інтерпретатора PHP та сервера баз даних MariaDB для подальшого розгортання та підтримки вебзастосунків.

Інструменти: гіпервізор VirtualBox; (за потреби) мережний емулятор GNS3; модель комп'ютерної мережі та структура домену, побудовані у межах попередніх лабораторних курсу, а також віртуальна машина з Ubuntu Server, налаштована у межах лабораторних робіт №19-20.

Теоретичні відомості та завдання до лабораторної роботи

ПРОГРАМНИЙ СТЕК LAMP ДЛЯ РОЗГОРТАННЯ ВЕБЗАСТОСУНКІВ

LAMP (Linux, Apache, MySQL/MariaDB, PHP) — один із найпоширеніших програмних стеків для розробки та експлуатації вебзастосунків. Стек являє собою набір взаємопов'язаних програмних компонентів, які забезпечують повноцінне серверне середовище для розміщення динамічних вебсайтів, вебпорталів та систем керування контентом.

Назва стеку утворена з перших літер його основних компонентів:

- **Linux** — операційна система, яка виступає платформою для роботи інших компонентів стеку;
- **Apache HTTP Server** — вебсервер, що приймає HTTP-запити від клієнтів та забезпечує доступ до вебресурсів;
- **MySQL або MariaDB** — система керування базами даних, призначена для зберігання та обробки інформації вебзастосунків;
- **PHP** — серверна мова програмування та інтерпретатор сценаріїв, за допомогою яких реалізується динамічна логіка вебзастосунків.

Протягом багатьох років стек LAMP залишається одним із найпопулярніших рішень для вебхостингу завдяки відкритості, високій надійності, широкій підтримці спільноти та значній кількості сумісного програмного забезпечення. На основі LAMP функціонують численні системи керування контентом, зокрема Drupal, WordPress та Joomla.

Історично літера **M** в абревіатурі LAMP означала систему керування базами даних MySQL. Згодом широкого поширення набула **MariaDB** — вільна та сумісна з MySQL система керування базами даних, створена розробниками оригінального проєкту MySQL. На сьогодні більшість дистрибутивів Linux, включаючи Ubuntu Server, використовують саме MariaDB як стандартну реалізацію SQL-сервера. Незважаючи на це, назва стеку LAMP залишається незмінною та може передбачати використання як MySQL, так і MariaDB.

У межах даної лабораторної роботи для реалізації компонента баз даних використовується **MariaDB**, оскільки вона входить до стандартних репозиторіїв Ubuntu, повністю підтримується сучасними вебзастосунками та забезпечує просте адміністрування і розгортання.

Важливим компонентом стеку є також **PHP**, параметри роботи якого визначаються конфігураційним файлом **php.ini**. За допомогою цього файлу адміністратор може налаштовувати обмеження на використання пам'яті, розмір файлів, що завантажуються,

часові параметри виконання сценаріїв та інші характеристики середовища виконання вебзастосунків.

У результаті розгортання стеку LAMP сервер Ubuntu отримує можливість обслуговувати динамічні вебзастосунки, які використовують бази даних та серверну логіку, що є необхідною умовою для подальшого встановлення і налаштування системи керування контентом Drupal.

Оновлення пакетів операційної системи

Перед встановленням програмного забезпечення доцільно оновити локальний кеш пакетів та встановити актуальні версії вже наявних компонентів системи.

Для цього виконайте команди:

sudo apt update

sudo apt upgrade -y

Команда apt update оновлює інформацію про доступні пакети в репозиторіях Ubuntu, а команда apt upgrade встановлює доступні оновлення програмного забезпечення.

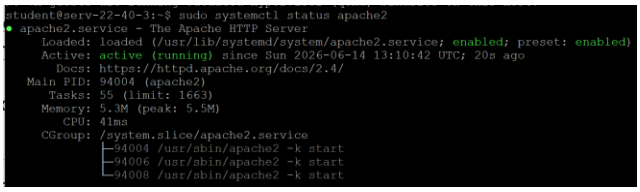
Встановлення та перевірка вебсервера Apache HTTP Server

Вебсервер Apache HTTP Server є одним із найпоширеніших вебсерверів у світі та входить до складу програмного стеку LAMP. Він забезпечує обробку HTTP-запитів клієнтів та надає доступ до вебресурсів, розміщених на сервері.

Для встановлення Apache виконайте команду:

sudo apt install apache2 -y

Після завершення встановлення перевіряємо стан служби:



sudo systemctl status apache2

У результаті виконання команди виведеться інформація про стан служби. Якщо встановлення пройшло успішно, серед параметрів буде присутній рядок *Active: active (running)*

Для перевірки автоматичного запуску служби під час завантаження операційної системи виконуємо:

sudo systemctl is-enabled apache2

У разі успішного налаштування буде отримано відповідь *enabled*

Також можна перевірити доступність вебсервера через браузер, відкривши адресу сервера *http://IP-адреса_сервера*

Якщо Apache працює коректно, відкриється стандартна сторінка привітання Ubuntu Apache2 Default Page.

Результат виконання цього кроку – встановлення та запуск вебсервера Apache HTTP Server, який готовий до подальшої інтеграції з PHP та сервером баз даних MariaDB у складі програмного стеку LAMP.

Завдання №21.1

1. Запустіть віртуальну машину з Ubuntu Server, налаштовану у межах лабораторних робіт №19 та 20. Увійдіть у систему під адміністративний обліковим записом, створеним під час встановлення системи.
2. Виконайте оновлення пакетів операційної системи та встановіть і перевірте роботу вебсервера Apache HTTP Server.

Встановлення сервера баз даних MariaDB

MariaDB є реляційною системою керування базами даних, яка використовується для зберігання та обробки даних вебзастосунків. Вона є сумісною з MySQL та входить до стандартних репозиторіїв Ubuntu Server, що спрощує її встановлення та адміністрування.

Для встановлення MariaDB виконуємо команду:

sudo apt install mariadb-server -y

Після завершення встановлення перевіряємо стан служби MariaDB:

sudo systemctl status mariadb

У разі успішного запуску служби серед параметрів буде присутній рядок Active: active (running). Також перевіряємо автоматичний запуск служби під час завантаження операційної системи:

sudo systemctl is-enabled mariadb

Очікуваний результат enabled.

Для перевірки встановленої версії сервера баз даних виконуємо команду:

mariadb --version

Приклад результату: *mariadb Ver 15.1 Distrib 10.11.x-MariaDB*

Після встановлення можна виконати підключення до сервера баз даних під обліковим записом адміністратора:

sudo mariadb

У разі успішного підключення буде відкрито консоль MariaDB: *MariaDB [(none)]>*

Для завершення роботи з консоллю виконайте команду:

exit;

Базове налаштування безпеки MariaDB

Після встановлення MariaDB рекомендується виконати початкове налаштування безпеки сервера баз даних. Для цього використовується утиліта mysql_secure_installation, яка дозволяє видалити тестові облікові записи та бази даних, а також обмежити можливість несанкціонованого доступу до сервера. Запускаємо утиліту налаштування:

sudo mysql_secure_installation



Під час виконання утиліта послідовно запропонує відповісти на низку запитань.

Налаштування автентифікації адміністратора

На сучасних версіях Ubuntu для локального адміністративного доступу використовується механізм автентифікації через обліковий запис операційної системи (unix_socket). Тому змінювати спосіб автентифікації не потрібно.

Switch to unix_socket authentication [Y/n]

Рекомендується залишити значення за замовчуванням: *Y*

Утиліта запропонує встановити пароль для облікового запису адміністратора БД.

Change the root password? [Y/n]

Для цілей лабораторної роботи доцільно залишити автентифікацію через системний обліковий запис та не встановлювати окремий пароль: *n*

Видалення анонімних користувачів

Remove anonymous users? [Y/n]

Рекомендується підтвердити видалення: *Y*

Анонімні користувачі можуть підключатися до сервера без автентифікації, що становить потенційну загрозу безпеці. Заборона віддаленого входу адміністратора

Disallow root login remotely? [Y/n]

Рекомендується підтвердити: *Y*

Адміністративний доступ до сервера баз даних повинен здійснюватися лише локально. Видалення тестової бази даних

Remove test database and access to it? [Y/n]

Рекомендується погодитися: *Y*

Тестова база даних не використовується в робочих середовищах і повинна бути видалена. Перезавантаження таблиць привілеїв

Reload privilege tables now? [Y/n]

Підтвердіть виконання: *Y*

Після завершення налаштування буде виведено повідомлення:

All done!

Перевірка доступу до сервера баз даних

Переконуємось, що підключення до MariaDB виконується коректно:

sudo mariadb

Після успішного входу відобразиться запрошення командного рядка MariaDB:

MariaDB [(none)]>

Для виходу з консолі використовуємо команду:

exit;

Таким чином, видалено анонімні облікові записи, заборонено віддалений доступ адміністратора, видалено тестову базу даних та оновлено таблиці привілеїв. Сервер баз даних підготовлено до використання у складі програмного стеку LAMP.

Завдання №21.2

1. Встановіть сервер баз даних MariaDB та перевірте його працездатність.
2. Виконайте початкове налаштування безпеки MariaDB за допомогою утиліти mysql_secure_installation.
3. Перевірте можливість локального адміністрування сервера баз даних через консоль MariaDB.

Встановлення PHP та модулів інтеграції з Apache і MariaDB

PHP є серверною мовою програмування, яка використовується для створення динамічних вебзастосунків. Під час обробки запиту вебсервер Apache передає PHP-сценарії інтерпретатору PHP, який виконує код і формує вебсторінку для клієнта.

Для встановлення PHP та необхідних модулів виконуємо команду:

sudo apt install php libapache2-mod-php php-mysql -y

де: *php* — основний інтерпретатор PHP;

libapache2-mod-php — модуль інтеграції PHP з вебсервером Apache;

php-mysql — модуль взаємодії PHP із серверами баз даних MySQL та MariaDB.

Після завершення встановлення перевіряємо версію PHP:

php -v

Приклад результату: *PHP 8.3.6 (cli)*

Переконуємося, що модуль PHP завантажений в Apache:

apache2ctl -M | grep php

Очікуваний результат: *php_module (shared)*



Рис.21.1. Версія та перевірка завантаження PHP

Для застосування змін перезапускаємо вебсервер Apache:

sudo systemctl restart apache2

Перевіряємо стан служби:

sudo systemctl status apache2

У разі успішного запуску буде відображено: *Active: active (running)*

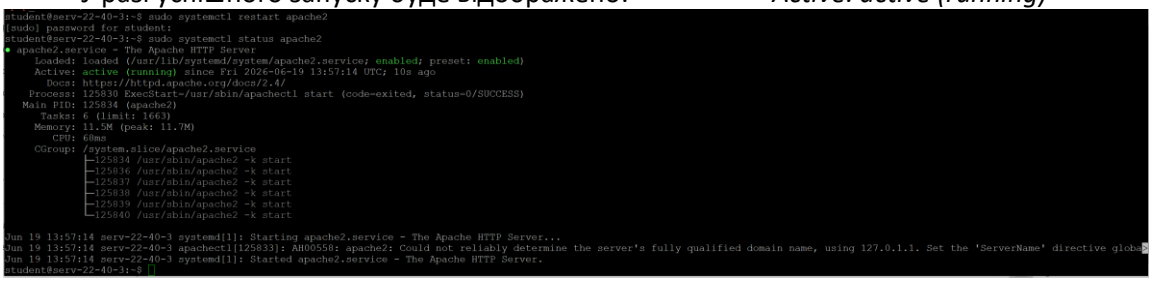


Рис.21.2. Перезавантаження PHP

Перевіряємо обробку PHP-сценаріїв. Створюємо тестовий файл у кореневому каталозі вебсервера:

sudo nano /var/www/html/info.php

Додаємо до файлу такий вміст:

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

Зберігаємо файл та відкриваємо у браузері адресу: *http://IP-адреса_сервера/info.php*

Якщо PHP працює коректно, буде відображено інформаційну сторінку з параметрами інтерпретатора PHP, завантаженими модулями та конфігурацією середовища виконання.

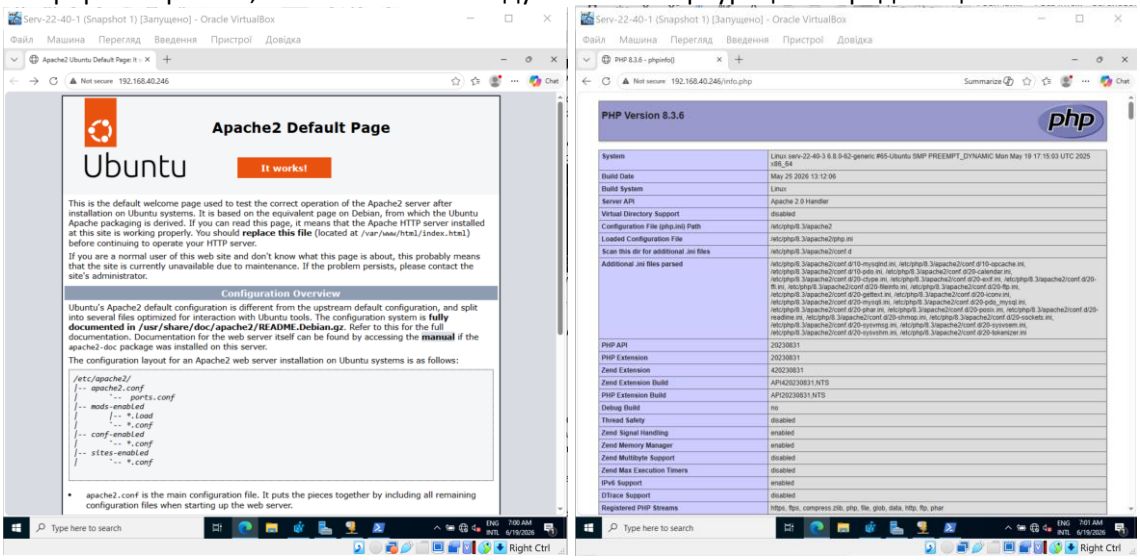


Рис.21.3. Перевірка обробки PHP-сценаріїв

Після завершення перевірки тестовий файл рекомендується видалити:

sudo rm /var/www/html/info.php

Таким чином на сервері встановлено PHP та модулі інтеграції з Apache і MariaDB. Вебсервер здатний виконувати PHP-сценарії та взаємодіяти із сервером баз даних, що забезпечує функціонування динамічних вебзастосунків.

Завдання №21.3

1. Встановіть інтерпретатор PHP та модулі, необхідні для взаємодії PHP із вебсервером Apache і сервером баз даних MariaDB.
2. Перевірте версію встановленого інтерпретатора та працездатність модуля PHP для Apache.
3. Створіть тестовий PHP-сценарій та перевірте його виконання через веббраузер.
4. Видаліть тестовий сценарій після завершення перевірки.

Встановлення додаткових модулів та налаштування середовища виконання PHP.

Більшість сучасних вебзастосунків та систем керування контентом використовують додаткові можливості PHP, які реалізуються у вигляді окремих модулів. Наприклад, Drupal потребує підтримки XML, багатобайтових кодувань, мережевих запитів та обробки графічних файлів.

Для встановлення рекомендованих модулів виконайте команду:

sudo apt install php-gd php-xml php-mbstring php-curl php-zip php-intl -y

де:

php-gd — забезпечує обробку графічних файлів та створення мініатюр зображень;

php-xml — забезпечує роботу з XML-документами;

php-mbstring — підтримує багатобайтові кодування символів;

php-curl — надає можливість виконання HTTP-запитів до зовнішніх ресурсів;

php-zip — забезпечує роботу з архівами ZIP;

php-intl — реалізує функції локалізації та міжнародних стандартів форматування даних.

Після завершення встановлення перевірте наявність встановлених модулів:

php -m

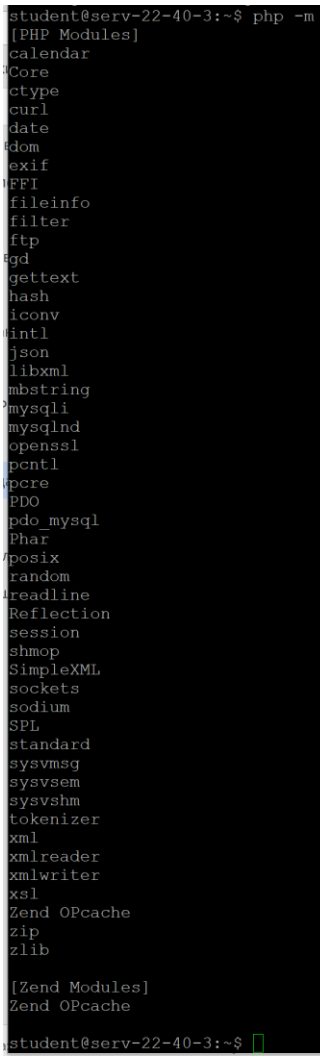
Для перегляду конкретного модуля можна використати команду:

php -m | grep mbstring

Аналогічно можна перевірити наявність інших модулів:

php -m | grep xml
php -m | grep curl
php -m | grep gd
php -m | grep zip
php -m | grep intl

У разі успішного встановлення буде виведено назву відповідного модуля.



Налаштування параметрів середовища PHP у файлі php.ini

Після встановлення інтерпретатора PHP та необхідних модулів доцільно виконати базове налаштування середовища виконання PHP. Основні параметри роботи інтерпретатора зберігаються у конфігураційному файлі php.ini.

Файл php.ini визначає правила роботи PHP та містить параметри, які впливають на використання оперативної пам'яті, час виконання сценаріїв, максимальний розмір файлів, що завантажуються, налаштування часових поясів та інші характеристики середовища

виконання. Для серверів Ubuntu з Apache конфігураційний файл PHP зазвичай розташований у каталозі */etc/php/8.x/apache2/php.ini*, де 8.x відповідає встановленій версії PHP. Актуальна на момент написання метлодикт версія php – 8.3, тому

sudo nano /etc/php/8.3/apache2/php.ini

Налаштування обсягу доступної пам'яті

Знайдіть параметр: *memory_limit = 128M*

та змініть його значення на: *memory_limit = 256M*

Параметр визначає максимальний обсяг оперативної пам'яті, який може використовувати один PHP-процес. Для сучасних CMS, зокрема Drupal, значення 256 МБ є достатнім для більшості завдань.

Налаштування максимального розміру завантажуваних файлів

Знайдіть параметр: *upload_max_filesize = 2M*

та змініть його на: *upload_max_filesize = 64M*

Параметр визначає максимальний розмір файла, який користувач може завантажити через вебінтерфейс.

Налаштування максимального розміру POST-запитів

Знайдіть параметр: *post_max_size = 8M*

та встановіть значення: *post_max_size = 64M*

Цей параметр визначає максимальний обсяг даних, які можуть передаватися серверу методом HTTP POST.

Налаштування часу виконання сценаріїв

Знайдіть параметр: *max_execution_time = 30*

та змініть його на: *max_execution_time = 300*

Параметр визначає максимальний час виконання PHP-сценарію в секундах. Збільшення цього значення дозволяє уникнути передчасного завершення тривалих операцій, наприклад встановлення модулів або імпорту даних.

Налаштування часової зони

Знайдіть рядок: *;date.timezone =*

та змініть його на: *date.timezone = Europe/Kyiv*

Цей параметр визначає часову зону, яка використовується PHP під час роботи з датою та часом.

Значення параметрів можуть відрізнятися залежно від версії PHP та вимог конкретного вебзастосунку, а наведені значення є рекомендованими для навчального стенда та більшості невеликих вебпроєктів.

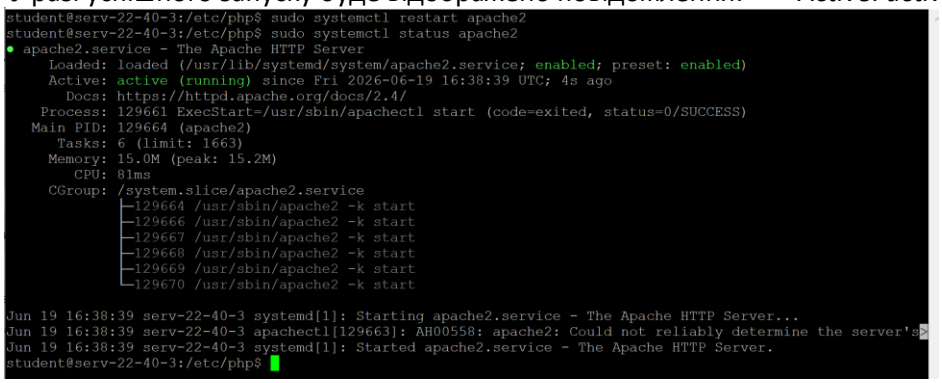
Після завершення редагування файла збережіть зміни та перезапустіть вебсервер Apache:

sudo systemctl restart apache2

Для перевірки відсутності помилок у конфігурації виконайте:

sudo systemctl status apache2

У разі успішного запуску буде відображено повідомлення: *Active: active (running)*



Завдання №21.4

1. Встановіть додаткові модулі PHP, необхідні для роботи сучасних вебзастосунків та систем керування контентом.
2. Перевірте наявність встановлених модулів у середовищі PHP.
3. Виконайте налаштування основних параметрів роботи PHP у файлі php.ini.
4. Перезапустіть вебсервер Apache та перевірте коректність застосування внесених змін.

Комплексна перевірка працездатності програмного стеку LAMP

Після встановлення та налаштування всіх компонентів програмного стеку необхідно переконатися у їхній спільній роботі. Для цього буде створено тестову базу даних та PHP-сценарій, який виконує підключення до сервера баз даних.

Створення тестової бази даних

Підключаємося до MariaDB:

sudo mariadb

Створюємо тестову базу даних:

CREATE DATABASE lamp_test;

Переконуємось у її наявності:

SHOW DATABASES;

У списку баз даних повинна відображатися база lamp_test.

```
student@serv-22-40-3:/etc/php$ sudo mariadb
[sudo] password for student:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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)]> CREATE DATABASE lamp_test;
Query OK, 1 row affected (0.009 sec)

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| lamp_test          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.011 sec)

MariaDB [(none)]> EXIT;
Bye
student@serv-22-40-3:/etc/php$
```

Рис. 21.4. Тестова БД

Створюємо окремого користувача для вебзастосунків:

CREATE USER 'lampuser'@'localhost' IDENTIFIED BY 'StrongPassword123!';

Надаємо права на тестову БД:

GRANT ALL PRIVILEGES ON lamp_test. TO 'lampuser'@'localhost';*

Застосовуємо зміни:

FLUSH PRIVILEGES;

Перевіряємо користувачів:

SELECT User, Host FROM mysql.user;

Виходимо:

EXIT;

```
student@serv-22-40-3:/etc/php$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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)]> CREATE USER 'lampuser'@'localhost' IDENTIFIED BY 'StrongPassword123!';
Query OK, 0 rows affected (0.012 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON lamp_test.* TO 'lampuser'@'localhost';
Query OK, 0 rows affected (0.010 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> SELECT User, Host FROM mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| lampuser    | localhost |
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
+-------------+-----------+
4 rows in set (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
student@serv-22-40-3:/etc/php$
```

Рис. 21.5. Користувач БД

Створення тестового PHP-сценарію

Створюємо файл тестового сценарію:

sudo nano /var/www/html/dbtest.php

Додаємо у нього наступний код:

```
<?php
$mysqli = new mysqli(
   "localhost",
   "lampuser",
   "StrongPassword123!",
   "lamp_test"
);
if ($mysqli->connect_error) {
   die("Помилка підключення: " . $mysqli->connect_error);
}
echo "Підключення до MariaDB успішно встановлено.";
$mysqli->close();
?>
```

Зберігаємо файл та закриваємо редактор.

Перевірка роботи стеку

У браузері відкриваємо адресу *http://IP-адреса_сервера/dbtest.php*

У разі успішної роботи всіх компонентів стеку буде відображено повідомлення:

Підключення до MariaDB успішно встановлено.

Отримання цього повідомлення свідчить про те, що вебсервер Apache коректно обробляє запити, PHP успішно виконує серверний код та має доступ до модуля роботи з MariaDB, а сервер баз даних MariaDB функціонує та приймає підключення.

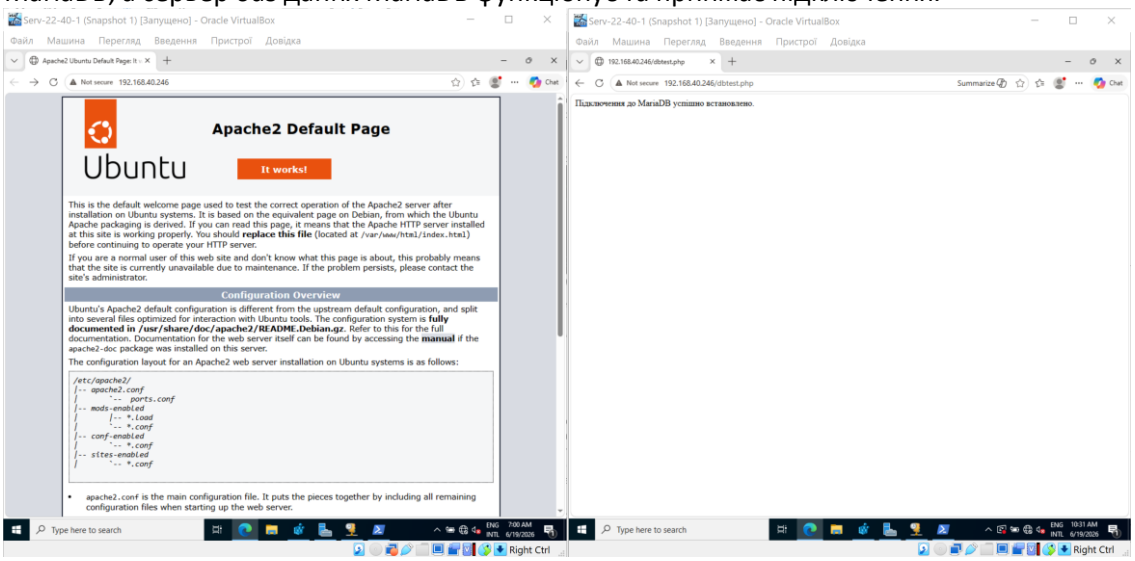


Рис.21.6. Перевірка обробки PHP-сценаріїв LAMP-стеком

Видалення тестових даних

Після завершення перевірки рекомендується видалити тестовий сценарій:

sudo rm /var/www/html/dbtest.php

За необхідності також можна видалити тестову базу даних:

```
sudo mariadb
DROP DATABASE lamp_test;
EXIT;
```

Таким чином виконано комплексну перевірку працездатності програмного стеку LAMP: підтверджено коректну взаємодію вебсервера Apache, інтерпретатора PHP та

сервера баз даних MariaDB. Серверне середовище готове до розгортання системи керування контентом Drupal у наступній лабораторній роботі.

Завдання №21.5

1. Створіть тестову базу даних у середовищі MariaDB для перевірки взаємодії компонентів програмного стеку LAMP.
2. Створіть окремого користувача бази даних та надайте йому необхідні привілеї для роботи з тестовою базою даних.
3. Виконайте перевірку підключення до створеної бази даних із використанням PHP-сценарію через вебсервер Apache.
4. Переконайтеся у коректній взаємодії компонентів стеку LAMP (Apache, PHP та MariaDB) шляхом успішного виконання тестового запиту.

Вимоги до звіту

Звіт має містити інформацію відповідно до завдань 21.1-21.5.

Рекомендовані джерела

- Network – Configuration – Ubuntu. URL: https://ubuntu.com/server/docs/network-configuration
- How To Install LAMP Stack (Apache, MySQL, PHP) on Ubuntu: https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu
- LAMP Stack Installation (…): Complete Production Guide: https://cubepath.com/docs/application-deployment/lamp-stack-installation-linux-apache-mysql-php