Лабораторна робота №23

Розгортання та налаштування розподіленого вебсередовища Drupal на базі окремих серверів вебслужб та баз даних

Мета: розгорнути та налаштувати багаторівневу архітектуру вебзастосунку Drupal на базі Ubuntu Server шляхом винесення сервера баз даних MariaDB на окрему віртуальну машину, забезпечивши взаємодію між вебсервером Apache, інтерпретатором PHP, системою керування контентом Drupal та віддаленим сервером баз даних.

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 №19-22.

Теоретичні відомості та завдання до лабораторної роботи

Загальні принципи багаторівневої архітектури вебзастосунків

Сучасні вебзастосунки, особливо системи керування контентом (CMS) та корпоративні інформаційні системи, у більшості випадків будуються на основі розподіленої багаторівневої архітектури. Такий підхід передбачає розділення функціональних компонентів системи між різними серверами або віртуальними машинами з метою підвищення продуктивності, масштабованості та надійності.

Найпоширенішою моделлю побудови вебсередовищ є трирівнева архітектура (3-tier architecture), яка включає:

- ✓ Рівень представлення (Presentation layer)
  вебсервер, який обробляє HTTP-запити користувачів та формує вебсторінки (у даній роботі — Apache HTTP Server).
- ✓ Рівень бізнес-логіки (Application layer)
  інтерпретатор серверної логіки (у даному випадку PHP та CMS Drupal), який реалізує обробку запитів і взаємодію з базою даних.
- ✓ Рівень зберігання даних (Data layer)
  сервер баз даних (MariaDB), який забезпечує централізоване зберігання та обробку структурованої інформації.

Розділення цих рівнів на різні фізичні або віртуальні вузли дозволяє реалізувати низку важливих переваг:

- ✓ Масштабованість
  кожен рівень можна масштабувати незалежно (наприклад, винести базу даних на окремий сервер).
- ✓ Підвищення продуктивності
  розподіл навантаження між вузлами зменшує навантаження на один сервер.
- ✓ Підвищення надійності
  відмова одного компонента не обов'язково призводить до повного виходу системи з ладу.
- ✓ Безпека
  розмежування доступу між рівнями дозволяє обмежити прямий доступ до критичних компонентів, зокрема бази даних.

У типовій реалізації такого підходу вебсервер взаємодіє з сервером баз даних через мережевий протокол MySQL/MariaDB, використовуючи TCP-з'єднання. При цьому база даних може розташовуватися на окремому вузлі, доступ до якого обмежується мережею та правилами автентифікації.

У рамках даної лабораторної роботи буде реалізовано розподілену конфігурацію вебсередовища Drupal, у якій:

- ✓ на першій віртуальній машині (VM1) розгорнуто вебсервер Apache, інтерпретатор PHP та CMS Drupal;
- ✓ на другій віртуальній машині (VM2) розгорнуто сервер баз даних MariaDB.

Такий підхід дозволяє отримати базове уявлення про принципи побудови реальних клієнт-серверних та багаторівневих вебсистем, які використовуються у промислових інформаційних середовищах.Початок форми

Клонування віртуальної машини вебсервера

Для побудови багаторівневої архітектури вебзастосунку необхідно створити окремий сервер баз даних. З метою скорочення часу розгортання використаємо механізм клонування існуючої віртуальної машини, на якій у попередній лабораторній роботі розгорнуто вебсередовище Drupal.

Завершуємо роботу операційної системи Ubuntu Server та вимикаємо віртуальну машину, на якій розгорнуто програмний стек LAMP і CMS Drupal.

Засобами системи віртуалізації (Oracle VirtualBox) створюємо повний клон віртуальної машини. Під час створення клону змінюємо ім'я повного клону та активуємо опцію повторної ініціалізації MAC-адрес мережевих адаптерів (Reinitialize the MAC address of all network cards або аналогічну).

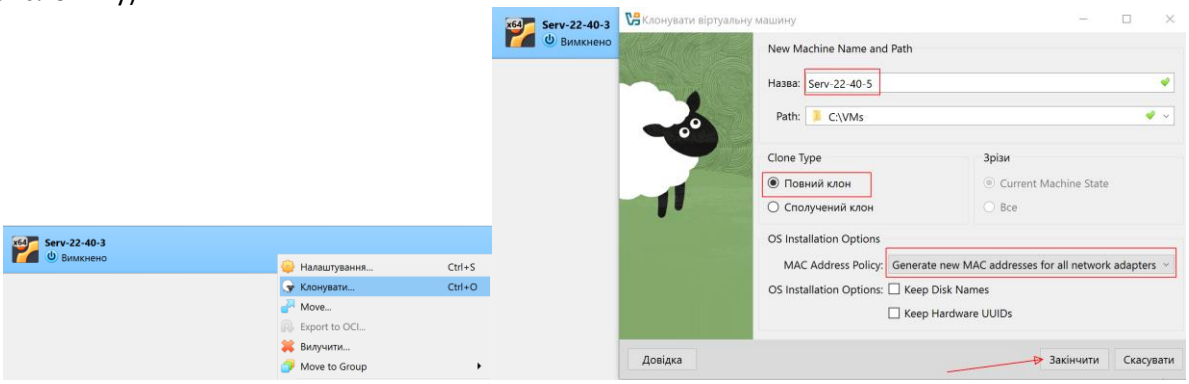


Рис.23.1. Клонування ВМ, на якій розгорнуто програмний стек LAMP і CMS Drupal.

Після завершення клонування переконуємось, що нова віртуальна машина має відмінну від оригіналу MAC-адресу мережевого адаптера.

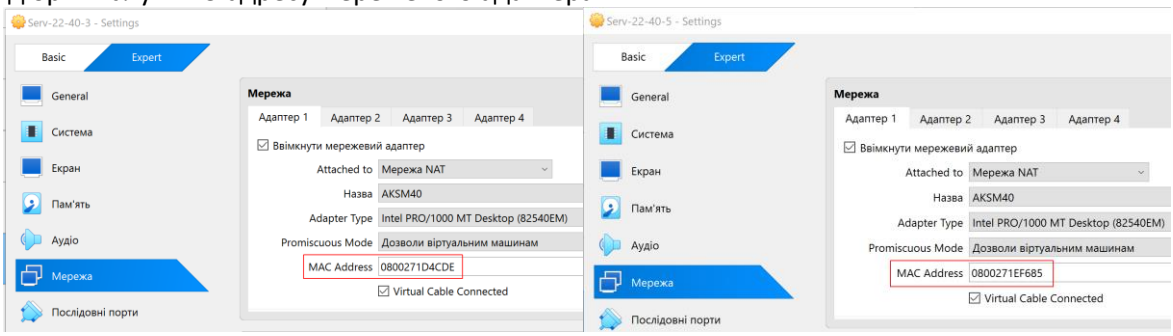


Рис.23.2. Порівняння MAC-адрес віртуальних машин.

Вмикаємо обидві віртуальні машини для подальшого налаштування та розподілу серверних ролей.

> ***Примітка.*** *На відміну від операційних систем сімейства Windows, Linux-системи не використовують унікальний ідентифікатор безпеки SID, тому клонування Linux-серверів зазвичай не потребує виконання додаткових процедур підготовки системи. Основними проблемами після клонування можуть бути дублювання MAC-адрес, IP-адрес, імен вузлів та інших мережевих параметрів, що може призвести до конфліктів у мережі.*

Завдання №23.1

1. Виконайте завершення роботи та вимкнення віртуальної машини, на якій розгорнуто програмний стек LAMP і CMS Drupal.
2. Створіть повний клон віртуальної машини засобами системи віртуалізації, забезпечивши генерацію нових MAC-адрес мережевих адаптерів. Ім’я клону має відповідати шаблону Serv-GG-VV-5, де GG - номер вашої групи, VV – ваш варіант.
3. Запустіть оригінальну та клоновану віртуальні машини та підготуйте їх до подальшого розподілу серверних ролей у багаторівневій архітектурі вебзастосунку.

Налаштування мережевих параметрів та імені клонованої віртуальної машини

Після клонування віртуальної машини оригінальний сервер та його копія матимуть однакові мережеві параметри та ім'я вузла. Одночасне використання двох ідентичних систем у мережі може призвести до конфліктів IP-адрес, некоректної роботи служб DNS, а також ускладнити адміністрування серверів.

Вмикаємо оригінальну та клоновану віртуальні машини. На клонованому сервері переглядаємо поточне ім'я вузла:

hostnamectl

Змінюємо ім'я клонованого сервера, наприклад, на Serv-GG-VV-5 :

sudo hostnamectl set-hostname Serv-22-40-5

Перевіряємо зміну імені вузла:

Hostnamectl

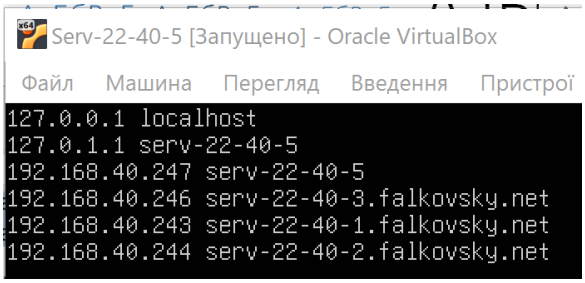
Редагуємо файл локального зіставлення імен вузлів:

sudo nano /etc/hosts

Замінюємо старе ім'я сервера на нове, наприклад:

127.0.1.1 Serv-22-40-5

Як



> ***Примітка.*** *У разі використання статичної IP-адресації необхідно змінити мережеві параметри клонованого сервера, призначивши йому унікальну IP-адресу відповідно до схеми адресації локальної мережі. Налаштування статичної IP-адреси виконується згідно з рекомендаціями, наведеними у додатку 23.1.*

Переконуємося, що оригінальна та клонована віртуальні машини використовують різні IP-адреси. Для перевірки мережевих параметрів виконуємо:

ip a

За потреби змінюємо параметри мережевого інтерфейсу клонованої віртуальної машини відповідно до схеми адресації локальної мережі. Після завершення налаштування виконуємо перезавантаження клонованого сервера:

sudo reboot

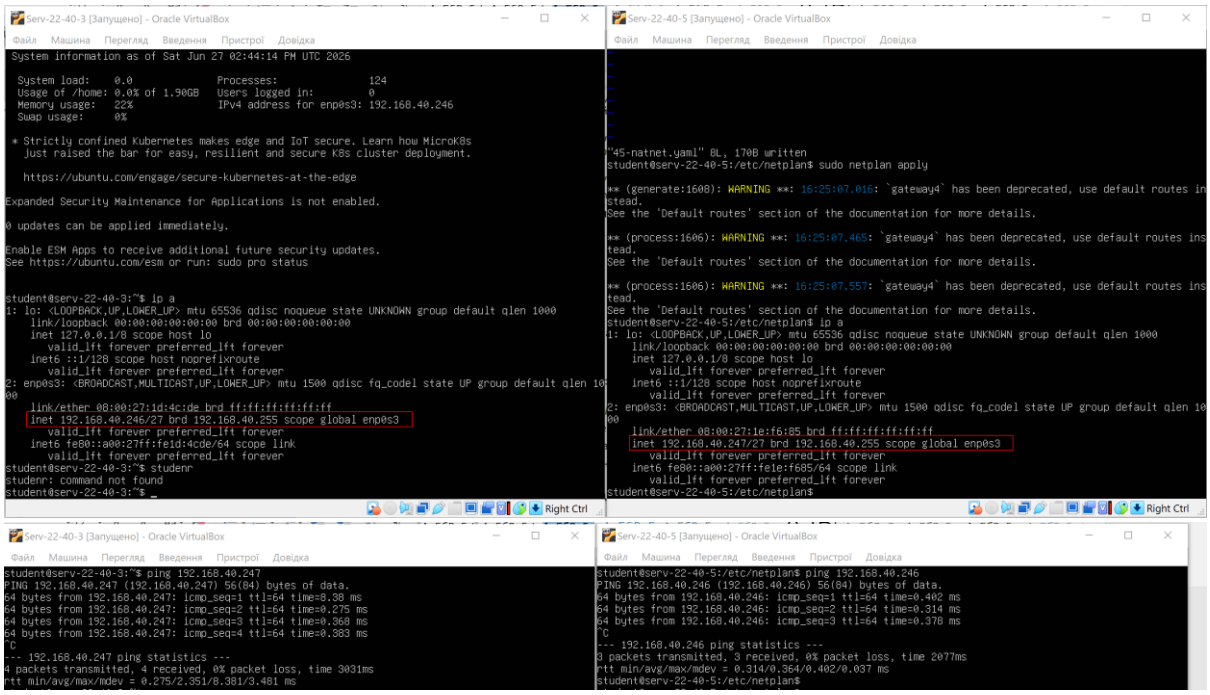


Рис.23.3. Перевірка мережевих налаштувань віртуальних машин.

> ***Примітка.*** *У межах лабораторної роботи імена серверів визначаються вимогами завдання. Проте під час експлуатації реальних інформаційних систем рекомендується використовувати зрозумілі та інформативні імена вузлів, що відображають їх функціональне призначення, наприклад:* ***web-server*** *— сервер вебзастосунку,* ***db-server*** *— сервер баз даних. Такий підхід спрощує адміністрування, моніторинг та супровід серверної інфраструктури.*

Завдання №23.2

1. Виконайте налаштування клонованої віртуальної машини, змінивши її ім'я вузла відповідно до ролі сервера у багаторівневій архітектурі вебзастосунку.
2. Перевірте та, за необхідності, змініть мережеві параметри клонованого сервера, забезпечивши унікальність його IP-адреси в локальній мережі.
3. Переконайтеся у відсутності конфліктів мережевих параметрів між оригінальною та клонованою віртуальними машинами.

Підготовка сервера баз даних до роботи у розподіленому вебсередовищі

Після клонування віртуальної машини на обох серверах присутні всі компоненти програмного стеку LAMP та CMS Drupal. На даному етапі клонована віртуальна машина буде використовуватися виключно як сервер баз даних, тому необхідно перевірити працездатність служби MariaDB та налаштувати можливість підключення до неї з віддаленого вебсервера.

На клонованому сервері перевіряємо стан служби MariaDB:

sudo systemctl status mariadb

Служба повинна перебувати у стані *active (running)*

Підключаємося до консолі MariaDB та переконуємося, що база даних Drupal, створена під час попередньої лабораторної роботи, присутня на сервері:

sudo mysql

SHOW DATABASES;

Для забезпечення доступу до сервера баз даних з вебсервера необхідно створити або модифікувати обліковий запис користувача MariaDB, дозволивши підключення з IP-адреси вебсервера. Наприклад, якщо вебсервер має IP-адресу 192.168.40.246, виконуємо:

CREATE USER 'drupal_user'@'192.168.40.246'
IDENTIFIED BY 'StrongPassword123!';

GRANT ALL PRIVILEGES ON drupal_db. TO*
'drupal_user'@'192.168.40.246';

FLUSH PRIVILEGES;

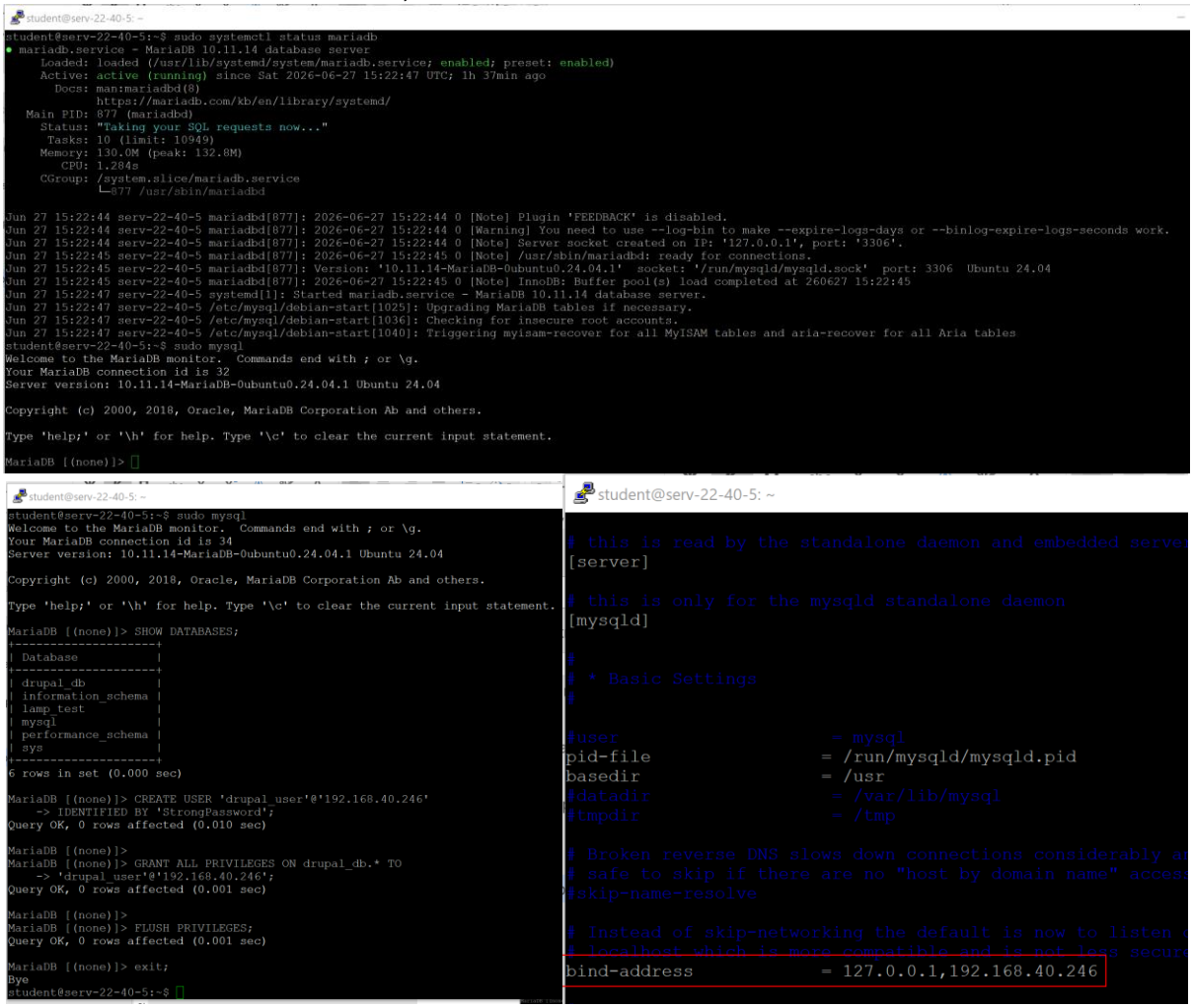


Рис.23.3. Створення та налаштування облікового запису 'drupal_user'.

Для забезпечення можливості віддаленого підключення перевіряємо параметр bind-address у конфігураційному файлі MariaDB:

sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

За замовчуванням параметр може мати вигляд:

bind-address = 127.0.0.1

Замінюємо його на нативну IP-адресу серверу бази даних WEB-стеку:

bind-address = 192.168.40.247

Після внесення змін перезапускаємо службу MariaDB:

sudo systemctl restart mariadb

Переконуємося, що сервер MariaDB прослуховує TCP-порт 3306:

sudo ss -tulnp | grep 3306

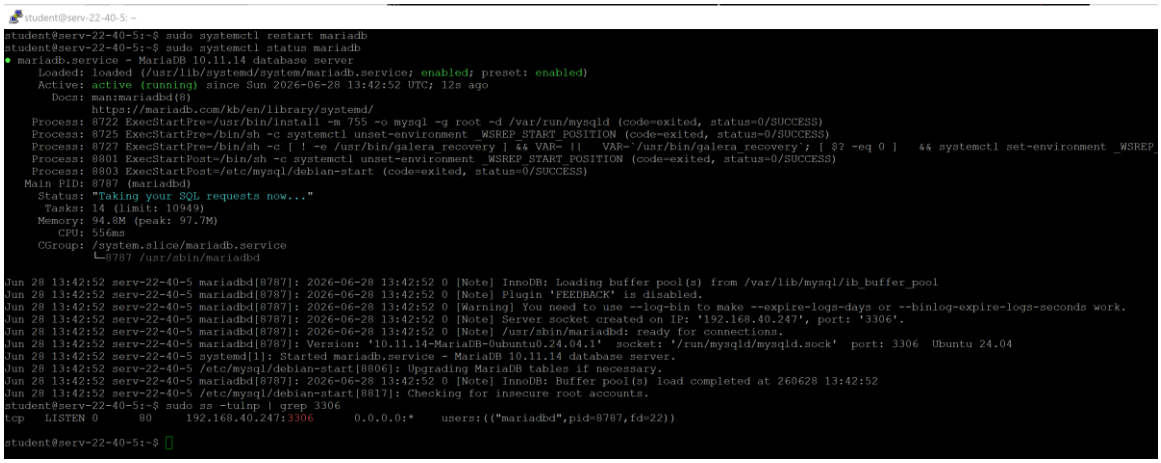


Рис.23.4. Перевірка налаштувань портів MariaDB на сервері Serv-22-40-5.

Завдання №23.3

1. Перевірте працездатність служби MariaDB на клонованій віртуальній машині та переконайтеся у наявності бази даних, створеної для CMS Drupal.
2. Створіть або модифікуйте обліковий запис користувача MariaDB для забезпечення віддаленого підключення з вебсервера, надавши йому необхідні привілеї до бази даних Drupal.
3. Налаштуйте параметр `bind-address` у конфігураційному файлі MariaDB для обмеження мережевих інтерфейсів, що використовуються для прийому підключень, відповідно до IP-адреси сервера баз даних.

Налаштування Drupal для роботи з віддаленим сервером баз даних

Після підготовки окремого сервера баз даних необхідно змінити параметри підключення CMS Drupal таким чином, щоб система використовувала віддалений сервер MariaDB.

На вебсервері (оригінальній віртуальній машині Serv-GG-VV-3) зупиняємо локальну службу MariaDB:

sudo systemctl stop mariadb

Переконуємося, що служба MariaDB зупинена:

sudo systemctl status mariadb

Стан служби повинен мати вигляд *inactive (dead)*

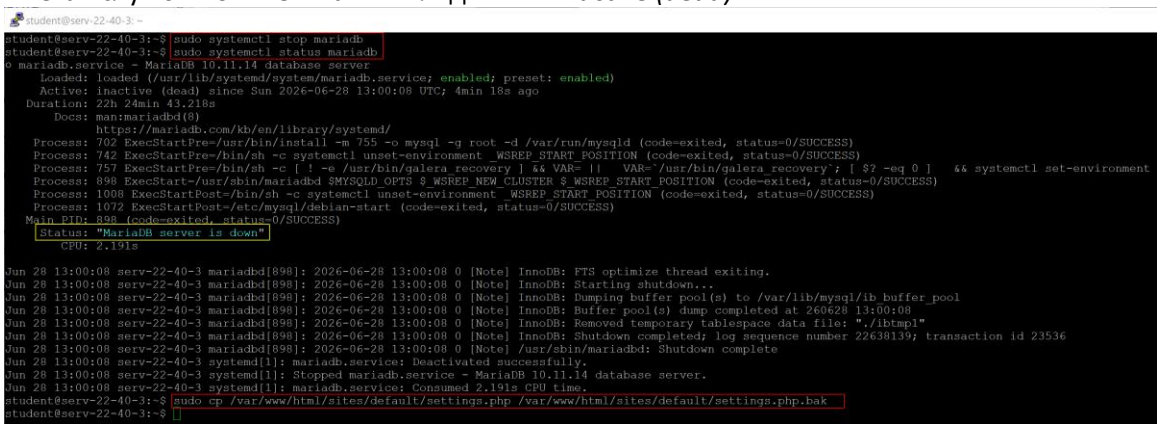


Рис.23.5. Зупинка MariaDB на Serv-22-40-3.

Створюємо резервну копію конфігураційного файлу Drupal:

sudo cp /var/www/html/sites/default/settings.php /var/www/html/sites/default/settings.php.bak

Відкриваємо файл конфігурації Drupal */var/www/html/sites/default/settings.php* для редагування та знаходимо секцію налаштування бази даних:

```
$databases['default']['default'] = array (
  'database' => 'drupal_db',
  'username' => 'drupal_user',
  'password' => 'StrongPassword123!',
  'prefix' => 'falkovsky',
  'host' => 'localhost',
  'port' => '3306',
  'isolation_level' => 'READ COMMITTED',
  'driver' => 'mysql',
  'namespace' => 'Drupal\\mysql\\Driver\\Database\\mysql',
  'autoload' => 'core/modules/mysql/src/Driver/Database/mysql/',
);
```

Замінюємо значення параметра host, вказавши IP-адресу сервера баз даних, наприклад:

'host' => '192.168.40.247',

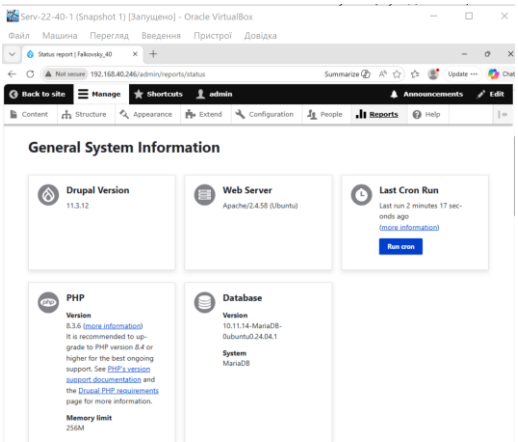
де 192.168.40.247 — IP-адреса клонованого сервера MariaDB. Зберігаємо внесені зміни та закриваємо редактор. Очищуємо кеш Drupal:

*sudo rm -rf /var/www/html/sites/default/files/php/**



Рис.23.6. Редагування параметру 'host' у файлі конфінурації /var/www/html/sites/default/settings.php.

Переходимо у браузері за адресою вебсайту Drupal та переконуємося у його працездатності.



Для підтвердження роботи з віддаленою базою даних тимчасово зупиніть службу MariaDB на сервері баз даних:

sudo systemctl stop mariadb

Після оновлення сторінки вебсайту Drupal повинен повідомити про помилку підключення до бази даних. Повторно запустіть службу:

sudo systemctl start mariadb

Якщо після запуску служби робота вебзастосунку відновлюється, це підтверджує успішне використання віддаленого сервера баз даних.

Завдання №23.4

1. Перевірте коректність роботи CMS Drupal після переналаштування на використання віддаленого сервера баз даних MariaDB через вебінтерфейс.
2. Переконайтеся у працездатності системи шляхом переходу до адміністративної панелі та перегляду системного звіту про стан (Status report).
3. Проаналізуйте конфігураційні параметри підключення до бази даних, використовуючи файл settings.php або механізм експорту конфігурації Drupal, та переконайтеся у відповідності налаштувань заданій архітектурі "WEB–DB".

Загальні положення до етапу очищення серверів

Після завершення розгортання та налаштування розподіленої архітектури вебзастосунку було отримано два ідентичні серверні середовища шляхом клонування віртуальної машини. У результаті на обох вузлах був присутній повний набір компонентів вебстеку (Apache, PHP, MariaDB та CMS Drupal).

Для забезпечення коректної багаторівневої архітектури типу “WEB–DB” необхідно виконати розподіл ролей серверів та видалити зайві компоненти, які не відповідають їх функціональному призначенню.

Даний етап виконується з метою:

- зменшення використання системних ресурсів;
- підвищення рівня безпеки за рахунок мінімізації встановленого програмного забезпечення;
- виключення потенційних конфліктів сервісів;
- приведення інфраструктури до типової промислової моделі розділення рівнів доступу.

Таким чином, вебсервер (Serv-GG-NN-3) повинен містити лише компоненти рівня представлення (Apache, PHP, Drupal), а сервер баз даних (Serv-GG-NN-5) — виключно СУБД MariaDB.

Очищення вебсервера (VM1 Serv-GG-NN-3) від локальної СУБД та зайвих компонентів

На вебсервері виконуємо видалення локального сервера баз даних MariaDB, оскільки доступ до БД здійснюється віддалено.

Зупиняємо службу MariaDB та вимикаємо автозапуск:

sudo systemctl stop mariadb

sudo systemctl disable mariadb

Видаляємо MariaDB (за потреби повного очищення) та очищаємо залишкові залежності

sudo apt remove --purge mariadb-server mariadb-client -y

sudo apt autoremove -y

Перевіряємо, що служба відсутня

systemctl status mariadb

Очікувано *Unit mariadb.service could not be found*

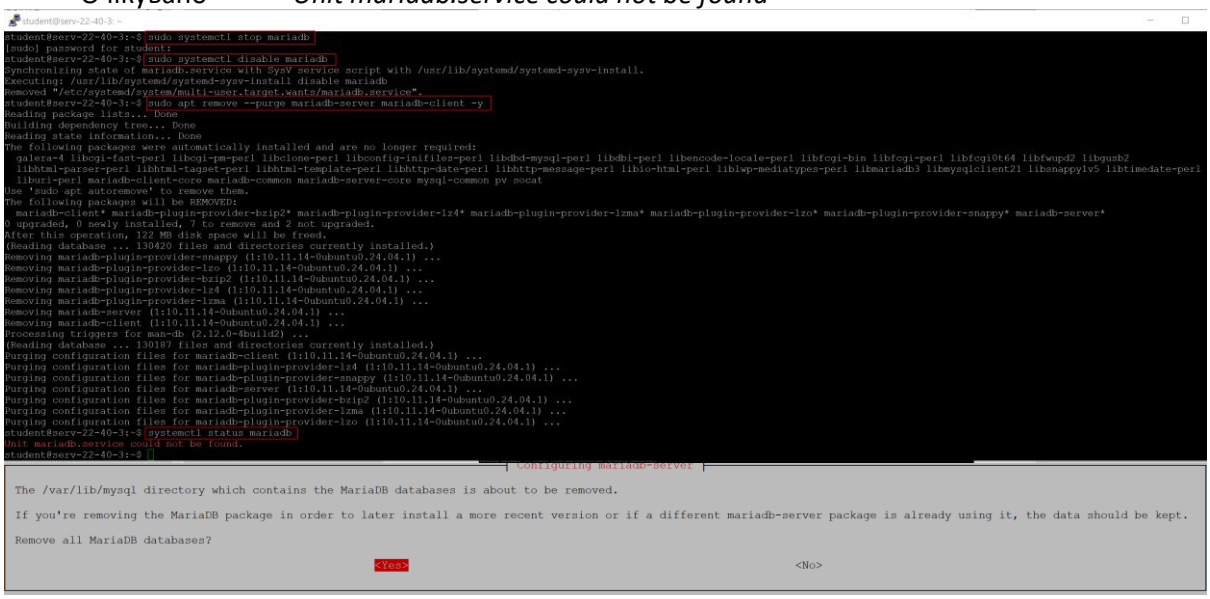


Рис.23.7. Видалення MariaDB на Serv-22-40-3

Очищення сервера баз даних (VM2 Serv-GG-NN-5) від вебкомпонентів

На сервері БД залишаємо лише MariaDB. Всі вебкомпоненти (Apache, PHP, Drupal) видаляються. Зупиняємо Apache, вимикаємо його автозапуск та видаляємо його:

sudo systemctl stop apache2
sudo systemctl disable apache2
sudo apt remove --purge apache2 -y

Видаляємо PHP та очищеємо залишки пакетів:

sudo apt remove --purge php -y*
sudo apt autoremove -y

Перевіряємо, що вебсервер відсутній

systemctl status apache2

Очікувано *Unit apache2.service could not be found*

Фінальна перевірка розподіленої архітектури

На WEB (Serv-GG-NN-3) має бути відсутній сервіс бази даних, а на DB (Serv-GG-NN-5) – мають бути відсутні сервіси WEB-стеку.

Виконуємо перевірку Drupal у браузері. Сайт має працювати у штатному режимі, використовуючи віддалену базу даних.

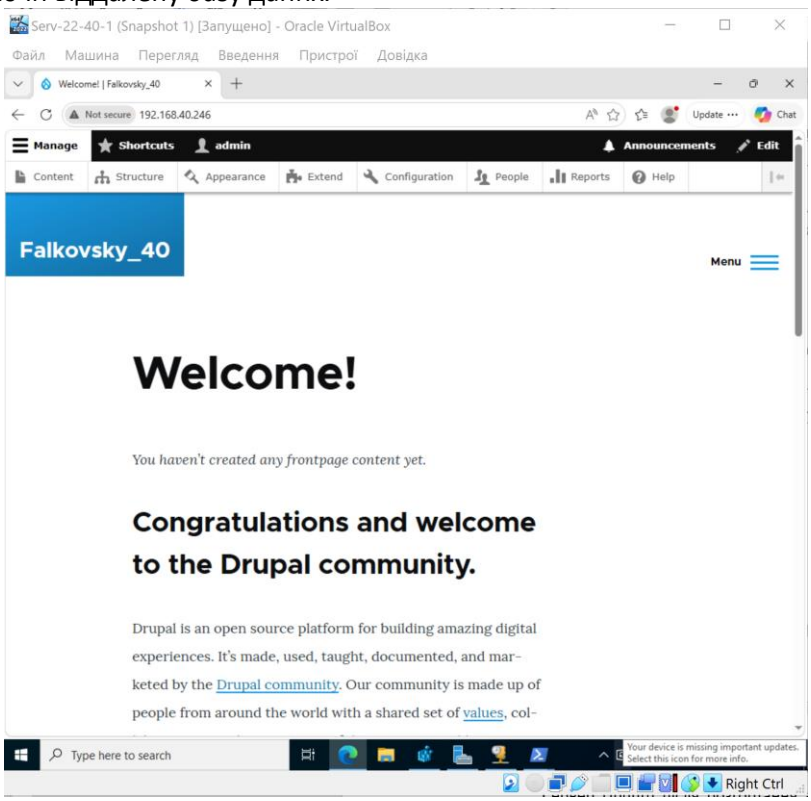


Рис.23.8. Фінальна перевірка працездатності розділеного WEB-стеку.

Завдання №23.5

1. Виконайте демонтаж компонентів вебстеку на сервері баз даних (Serv-GG-NN-5), залишивши лише сервер MariaDB для забезпечення його роботу як окремого рівня зберігання даних.
2. Видаліть сервер баз даних MariaDB на вебсервері (Serv-GG-NN-3), залишивши лише вебкомпоненти (Apache, PHP, Drupal).
3. Перевірте працездатність вебзастосунку Drupal після розділення ролей серверів та переконайтесь у коректній взаємодії між WEB- та DB-рівнями.

Вимоги до звіту

Звіт має містити інформацію відповідно до завдань 23.1-23.5.

Додаток 23.1.

Налаштування статичної адреси для Ubuntu-server

Сервер Ubuntu після розгортання зконфігуровано на динамічну адресацію. Сервери, зазвичай, працюють на статиці, тому необхідно виконати зміну налаштування динамічної адреси на статичну. Шукаємо назву мережевого інтерфейсу, який необхідно відредагувати ***ip a***

Шукаємо конфігураційні файли Netplan (з розширенням YAML), що зберігаються в каталозі /etc/netplan. Ймовірно, у цьому каталозі є один або декілька файлів YAML. Назва файлу може відрізнятися залежно від налаштувань та версії операційної системи.

sudo dir /etc/netplan

Наприклад, у каталозі знайдено файл 50-cloud-init.yaml. Робимо його копію для редагування:

sudo cp /etc/netplan/50-cloud-init.yaml /etc/netplan/40-natnet.yaml

Відкриваємо для редагування

sudo vi /etc/netplan/40-natnet.yaml

Нижче наведено вміст файлу Netplan до та після редагування.

/etc/netplan/50-cloud-init.yaml --- DHCP

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
```

/etc/netplan/40-natnet.yaml Static 192.168.40.246/27

```
network:
  version: 2
  ethernets:
    enp0s3:
      addresses: [192.168.40.246/27]
      gateway4: 192.168.40.241
      nameservers:
        addresses: [192.168.40.243]
```

Зберігаємо відредагований Netplan та застосовуємо внесені зміни:

sudo netplan apply

Помилки у файлі Netplan можуть бути локалізовані за допомогою інструменту yamllint. Встановлення та приклад застосування:

sudo apt install yamllint
yamllint /etc/netplan/40-natnet.yaml

Перевірка ip a покаже, що на інтерфейсі enp0s3 активні дві одночасні IPv4-адреси:

192.168.40.246/27— ручна статична адреса, задана через netplan;

192.168.40.xxx/27— динамічна (DHCP) адреса, отримана автоматично, і позначена як secondary dynamic.

Щоб не отримувати динамічну адресу, перейменовуємо (приховуємо ☺) цю конфігурацію та застосовуємо зміни:

sudo mv /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.bkp
sudo netplan apply

Рекомендовані джерела

- 🎧 Drupal:
  https://new.drupal.org/home
- 🎧 Wikipedia. Drupal:
  https://uk.wikipedia.org/wiki/Drupal
- 🎧 Drupal in Ukraine:
  https://drupal.org.ua/en
- 🎧 Drupal recommends MariaDB:
  https://mariadb.org/drupal-recommends-mariadb/