Лабораторна робота №6

Навчальна дисципліна

Інженерія надійності програмного забезпечення

Виконав студент 544 групи

Максимович Микола Юрійович

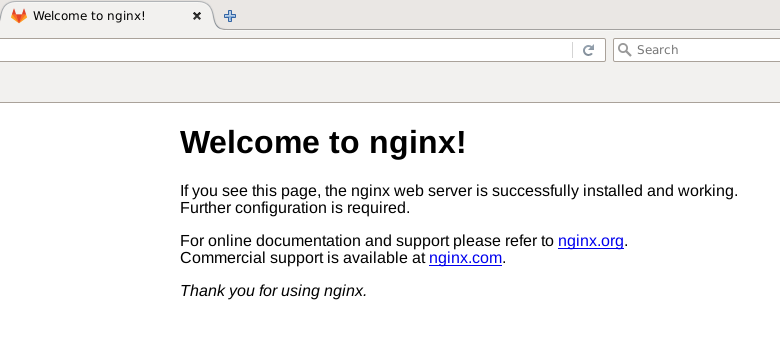
**Завдання до лабораторної роботи**

Розгорнути web-застосунок у контейнері та зробити навантаження на нього. Подати результати виконання у вигляді звіту.

**Виконання лабораторної роботи**

Інсталювавши необхідне ПЗ, а саме: Apache Benchmark, Docker, та налаштувавши його я запустив Докер контейнер з образом web-серверу NGINX задавши порт 8080.

Щоб перевірити чи все працює потрібно через браузер звернутись до локалхосту з портом 8080:

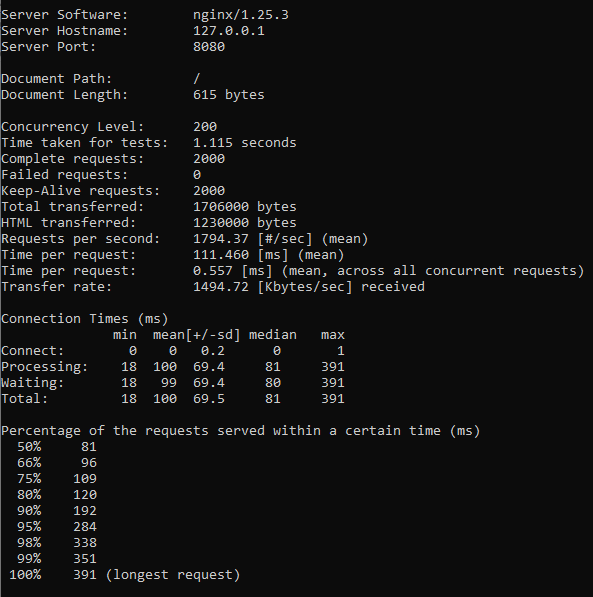


У відповідь було отримано стартові сторінку з якої можна зрозуміти що застосунок запущений та готовий до роботи.

Далі за допомогою команди:

$ ab -n 2000 -c 200 -k <http://127.0.0.1:8080/>

Я запускаю навантаження на розгорнутий застосунок. А результат наведено нижче на скріншоті:



Можна побачити, що тестування було проведено з **2000** одночасних з’єднань з сервером, і час, необхідний для завершення тестування, становив **1.115 секунд**. Загалом було виконано **2000** запитів, і всі вони були успішними. Також можна побачити детальну інформацію про тестування, таку як час з’єднання, довжину документа, швидкість передачі даних, тощо.

**Контрольні запитання**

1. Docker - це інструмент для запуску ваших додатків у контейнерах. Контейнери пакують всі залежності та код, який вашому додатку потрібен для роботи, у один файл, який буде працювати однаково на будь-якій машині. Docker схожий на концепцію віртуальних машин, але набагато легший. Замість запуску повноцінної окремої операційної системи (що є великим навантаженням), Docker запускає контейнери, які використовують ту ж хост-операційну систему і віртуалізуються лише на рівні програмного забезпечення.
2. Apache Benchmark (ab) - це інструмент для оцінки продуктивності веб-сервера шляхом засипання його HTTP-запитами та запису метрик затримки та успіху. Apache Benchmark може допомогти вам визначити, скільки трафіку ваш HTTP-сервер може витримати, перш ніж продуктивність почне погіршуватися, і встановити базові значення для типових часів відгуку.
3. HTTP (HyperText Transfer Protocol) - це протокол, який використовується для доступу до даних у Всесвітньому вебі (www). Протокол HTTP може використовуватися для передачі даних у вигляді простого тексту, гіпертексту, аудіо, відео тощо. HTTP - це протокол клієнт-сервер: запити ініціюються отримувачем, зазвичай веб-браузером. Повний документ відтворюється з різних піддокументів, отриманих, наприклад, тексту, опису макета, зображень, відео, сценаріїв та іншого.
4. Не кожен веб-застосунок має використовувати протокол HTTP. Використання стандартного HTTP, як правило, не створює проблем, якщо текст містить лише загальну інформацію, наприклад, для завантаження публічної веб-сторінки. Однак, якщо він містить конфіденційні дані, такі як імена користувачів, паролі або дані кредитних карток, використання незашифрованого HTTP може створити серйозні проблеми з безпекою5. Крім того, є інші протоколи 7-го рівня моделі OSI, які можуть бути використані в залежності від вимог до застосунку.
5. Тестування на навантаження - це тип тестування продуктивності, який оцінює продуктивність системи, програмного продукту або програмного застосунку в умовах реального навантаження. Основна мета тестування на навантаження - зрозуміти, як система обробляє користувацький трафік та транзакції, забезпечуючи її стабільність та доступність при таких умовах. Під час тестування на навантаження моделюються різні сценарії для тестування поведінки системи при різних умовах навантаження. За результатами тестування на навантаження можна отримати інформацію про максимальну кількість користувачів або транзакцій, які система може обробити, а також виявити та вирішити будь-які проблеми продуктивності.