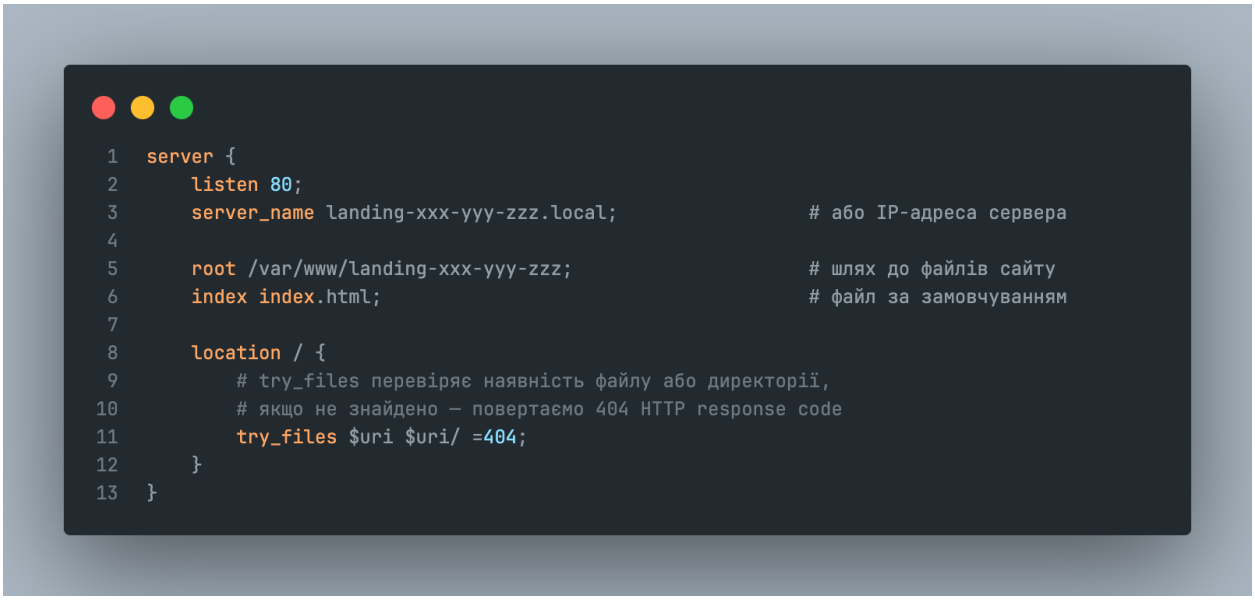
**Практична робота №2**

# ДОСЛІДЖЕННЯ РОБОТИ З ДИРЕКТИВАМИ В NGINX. СТВОРЕННЯ ТА БАЗОВЕ НАЛАШТУВАННЯ САЙТІВ

*Мета заняття*: дослідити структуру конфігураційного файлу Nginx та вкладеність контекстів; навчитися налаштовувати робочі процеси та параметри продуктивності; зрозуміти механізм наслідування директив між контекстами; набути практичних навичок роботи з директивою include; створювати сайти у Nginx для тестування різних сценаріїв роботи вебсерверу.

## Завдання на роботу

1. Проаналізувати головний конфігураційний файл Nginx `/etc/nginx/nginx.conf`. Описати вміст конфігурації у форматі списку чи таблиці.
2. Створити статичний сайт-візитку.
    a. Створити директорію `/var/www/landing-xxx-yyy-zzz`, де `xxx` – назва групи (наприклад, kim251), `yyy` – номер варіанту в групі (наприклад, для 5 варіанту це буде 005), `zzz` – ініціали студента (наприклад, mvv);
    b. Перевірити права на створену директорію, отримати інформацію про користувача та групи власника;
    c. Створити файл `/var/www/landing-xxx-yyy-zzz/index.html`. Редагувати його відповідно до HTML-розмітки. У файлі може міститись, наприклад, інформація про Вас.
    d. Створити сайт в sites-available: `/etc/nginx/sites-available/landing-xxx-yyy-zzz` з наступним вмістом (див.рис.2.d):



```
server {
    listen 80;
    server_name landing-xxx-yyy-zzz.local;                # або IP-адреса сервера

    root /var/www/landing-xxx-yyy-zzz;                    # шлях до файлів сайту
    index index.html;                                     # файл за замовчуванням

    location / {
        # try_files перевіряє наявність файлу або директорії,
        # якщо не знайдено — повертаємо 404 HTTP response code
        try_files $uri $uri/ =404;
    }
}
```

Рис. 2.d – Сайт в `sites-available`

e. Активувати конфігурацію сайту, створивши символічне посилання, виконавши команду `sudo -ln -s /etc/sites-available/landing-xxx-yyy-zzz /etc/nginx/sites-enabled/`
f. Виконати перевірку синтаксису Nginx та виправити помилки за їх наявності;
g. Перезавантажити конфігурацію Nginx;
h. Перевірити доступність сайту, виконавши наступну команду: `curl --resolve landing-xxx-yyy-zzz.local:80:127.0.0.1 http://landing-xxx-yyy-zzz.local`

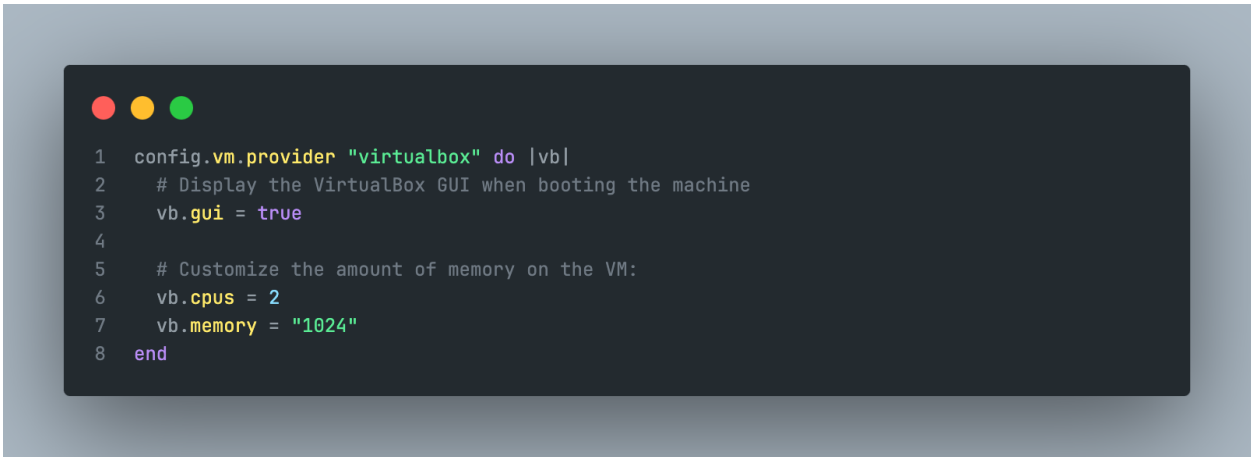
i. Перевірити як Nginx повертає сторінку з помилкою 404 Not Found, виконавши наступну команду: `curl --resolve landing-xxx-yyy-zzz.local:80:127.0.0.1 http://landing-xxx-yyy-zzz.local$RANDOM`

3. Робота з worker-процесами.

a. Перевірити значення значення worker_processes у `/etc/nginx/nginx.conf`;

b. Перевірити поточну кількість worker-процесів;

c. Вийти з ssh сесії на віртуальній машині та зупинити її, використовуючи відповідну команду для vagrant. Змінити кількість ядер віртуальної машини, редагувавши Vagrantfile, як показано на рис. 3.d. Встановити значення `vb.cpus = 1`. Запустити віртуальну машину та підключитись по ssh до неї, використовуючи відповідні команди vagrant.



```
config.vm.provider "virtualbox" do |vb|
  # Display the VirtualBox GUI when booting the machine
  vb.gui = true

  # Customize the amount of memory on the VM:
  vb.cpus = 2
  vb.memory = "1024"
end
```

Рис. 3.d – Конфігурація Vagrantfile для зміни кількості vCPU

d. Перевірити кількість worker-процесів після проведених маніпуляцій;

e. Відкотити попередні зміни та переконатись, що кількість worker-процесів така ж, як і була до здійснених дій.

f. Встановити кількість worker-процесів вручну, щоб вона не залежала від кількості ядер. Задати значення `worker_processes 1;` у `/etc/nginx/nginx.conf`, перевірити справність конфігурації та надіслати сигнал для graceful reload конфігурації для master-процесу nginx;
g. Перевірити кількість worker-процесів після проведених маніпуляцій;
h. Відкотити попередні зміни, щоб кількість worker-процесів знову визначалась автоматично та залежала від кількості ядер.

4. Налаштувати worker connections.
    a. Визначити поточне значення worker connections, яке встановлене у `/etc/nginx/nginx.conf`;
    b. Змінити поточне значення на 384, перевірити справність конфігурації та надіслати сигнал для graceful reload конфігурації для master-процесу nginx;
    c. Порахувати теоретичну максимальну кількість одночасних з'єднань до сервера;
    d. Відкотити попередні зміни та повернути значення для worker connections.
5. Дослідження механізму наслідування директив.
    a. Створити новий сайт, що буде повертати системну інформацію про систему, на якій запущено Nginx, створивши файл `/etc/nginx/sites-available/sysinfo-xxx-yyy-zzz`.

b. Додати у цей сайт наступний вміст:

```
server {
    listen 80;
    server_name sysinfo-xxx-yyy-zzz.local;

    add_header X-Inherited-From "server" always;

    default_type application/json;

    location = /hostname {
        return 200 '{"hostname":"$hostname"}';
    }

    location = /whoami {
        add_header X-Inherited-From "location:/whoami" always;
        return 200 '{"remote_user":"$remote_user"}';
    }

    location = /id {
        add_header X-Inherited-From "location:/id" always;
        return 200 '{"worker_pid":"$pid"}';
    }

    location = /serverinfo {
        return 200
        '{
        "hostname":"$hostname",
        "server_name":"$server_name",
        "pid":"$pid",
        "remote_addr":"$remote_addr",
        "request_uri":"$request_uri"
        }';
    }
}
```

Рис. 4.b – Конфігурація для системного сайту

c. Активувати конфігурацію сайту, створивши символічне посилання, виконавши команду `sudo -ln -s /etc/sites-`

`available/sysinfo-xxx-yyy-zzz /etc/nginx/sites-enabled/`

d. Виконати перевірку синтаксису Nginx та виправити помилки за їх наявності;

e. Перезавантажити конфігурацію Nginx;

f. Проаналізувати успадковування через звертання до усіх URI, використовуючи curl.

6. Створити сайт для тесту сніпетів стиснення та хедерів безпеки.

a. Створити директорії для наступної структури сайту, що представлена на рис.6.a. використовуючи наступну команду:
`sudo mkdir -p /var/www/security-and-compression-xxx-yyy-zzz/{css,js,data}`


Рис.6.a – Структура сайту

b. Створити наступні файли для сайту (див рис.6.b.1 – 6.b.5):


Рис.6.b.1 – Вміст index.html


Рис.6.b.2 – Вміст style.css


Рис.6.b.3 – Вміст app.js

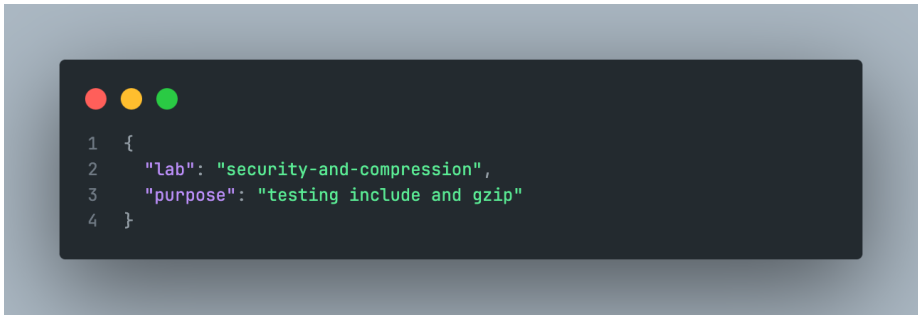



Рис.6.b.4 – Вміст sample.json


Рис.6.b.5 – Вміст image.svg

c. Виконати команду для генерації вмісту текстового файлу: `yes "Testing gzip compression line." | head -n 500 > \ /var/www/security-and-compression-xxx-yyy-zzz/large.txt`

d. Створити сніпет для конфігурації стиснення `/etc/nginx/snippets/gzip.conf` з вмістом, що пресдтавлено на рис. 6.d:


Рис.6.d – Вміст gzip.conf

e. Створити `/etc/nginx/snippets/security-headers.conf`, в якому будуть розміщені заголовки безпеки. Файл матиме вміст, представлений на рис.6.e:

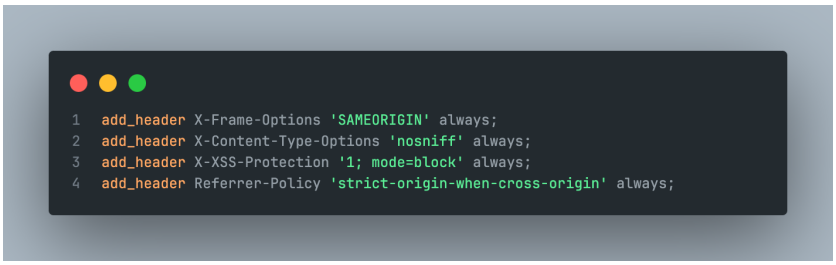



Рис.6.e – Вміст security-headers.conf

f. Підключити gzip.conf в nginx.conf;

g. Створити сайт `security-and-compression-xxx-yyy-zzz` в `/etc/nginx/sites-available` з вмістом, зображеним на рис.6.g:


Рис.6.e – Вміст security-and-compression-xxx-yyy-zzz

h. Перевірити справність конфігурації та надіслати сигнал для graceful reload конфігурації для master-процесу nginx;
i. Перевірити застосування security headers та gzip, проаналізувавши response header, використовуючи curl.

7. Вийти з SSH-сесії, ввівши команду `exit` та зупинити віртуальну машину, виконавши команду `vagrant halt`. Надати скріншот результату.

## Контрольні запитання

1. Яку роль відіграє центральний конфігураційний файл вебсервера у формуванні загальної структури налаштувань?
2. У чому полягає принцип модульності конфігурації та навіщо розділяти її на окремі файли?
3. Яке призначення механізму активації віртуальних хостів через символічні посилання?
4. Як відбувається вибір відповідного server-блоку при обробці HTTP-запиту?
5. У чому полягає різниця між перевіркою конфігурації та її застосуванням без перезапуску сервісу?
6. Як кількість процесів сервера пов’язана з апаратними ресурсами системи?
7. Яким чином ліміти операційної системи впливають на максимальну кількість одночасних з’єднань?
8. Що таке наслідування директив у конфігурації веб-сервера та як працює їх перевизначення на нижчих рівнях?
9. Які переваги надає винесення спільних налаштувань (наприклад, безпеки чи стиснення) в окремі конфігураційні фрагменти?
10. Які інструменти можна використати для тестування HTTP-поведінки сервера та аналізу його відповіді?