**Звіт**

Тема: Робота з SSH та Nginx

КИЇВ – 2023

**Зміст**

|  |  |
| --- | --- |
| **Вступ…………………………………………………………………………** | **3** |
| **Хід роботи…………………………………………………………………...** | **5** |
| **Встановлення і налаштування SHH…………………………………….** | **5** |
| **Встановлення і налаштування Nginx…………………………………...** | **9** |
| **Висновок…………………………………………………………………….** | **15** |

**Вступ**

**SSH (Secure Shell)** - це криптографічний мережевий протокол, який забезпечує безпечний доступ до віддаленого комп'ютера через незахищену мережу, таку як Інтернет. В основі протоколу SSH лежить шифрування, що робить його відповідним для захищеного передавання даних і автентифікації користувачів.

Основна суть SSH полягає в тому, що він дозволяє здійснювати віддалене керування комп'ютером або сервером з будь-якого іншого пристрою, який підтримує SSH-протокол. Ви можете віддалено виконувати команди на віддаленому комп'ютері, передавати файли між локальним і віддаленим пристроями, встановлювати і налаштовувати програмне забезпечення і багато іншого.

Ось деякі переваги та можливості SSH:

* Безпека: SSH надає шифрування трафіку, що захищає дані від перехоплення або зламу. Він також має механізми аутентифікації, які забезпечують, що тільки правильні користувачі можуть отримати доступ.
* Віддалений доступ: SSH дозволяє вам віддалено управляти серверами або комп'ютерами без необхідності фізичного присутності. Це дозволяє адміністраторам систем і розробникам здійснювати керування та налагодження на віддалених машинах зручним способом.
* Передача файлів: SSH має вбудовану підтримку для передачі файлів між локальним і віддаленим комп'ютерами за допомогою команди scp (Secure Copy) або за допомогою програми для клієнта SSH, яка має графічний інтерфейс.
* Тунелювання портів: SSH може створювати зашифровані тунелі, що дозволяють вам безпечно передавати трафік між віддаленими машинами через незахищені мережі. Це може бути корисно для доступу до внутрішніх ресурсів мережі або для захисту інформації, переданої між віддаленими машинами.
* Керування ключами: SSH використовує ключі для аутентифікації замість паролів. Ви можете створити публічний і приватний ключі SSH на своєму локальному комп'ютері і додати публічний ключ до віддаленого сервера. Це зробить процес аутентифікації більш безпечним і зручним, оскільки вам не потрібно буде вводити пароль кожного разу при підключенні.
* Прикладний програмний інтерфейс (API): SSH має API, який дозволяє розробникам використовувати функціонал SSH у своїх програмах. Це дає можливість автоматизувати взаємодію з віддаленими серверами, створювати сценарії автоматичного резервного копіювання, налагодження та інші завдання.
* Переносимість: Протокол SSH є стандартом для безпечного віддаленого доступу і підтримується більшістю операційних систем, включаючи Linux, macOS і Windows. Це означає, що ви можете використовувати SSH для взаємодії з різними системами без необхідності встановлення додаткових програм або протоколів.

SSH є надійним і безпечним протоколом для віддаленого доступу і керування комп'ютерами та серверами. Його шифрування трафіку та механізми аутентифікації забезпечують конфіденційність і цілісність даних, а також запобігають несанкціонованому доступу. Узагальнюючи, SSH є потужним і безпечним інструментом для забезпечення захищеного доступу та керування віддаленими комп'ютерами і серверами, що робить його незамінним для системних адміністраторів, розробників та користувачів, які потребують віддаленого доступу до ресурсів.

**Nginx** — вільний вебсервер і проксі-сервер. Є версії для сімейства Unix-подібних операційних систем (FreeBSD, GNU/Linux, Solaris, Mac OS X) та Microsoft Windows.

На відміну від інших продуктів даного сегмента Nginx використовує інший принцип обробки вхідних даних. ПЗ розбиває кожен запит користувача на кілька дрібних, спрощуючи таким чином обробку кожного. У термінології Nginx вони отримали назву робоче з'єднання. Після обробки кожне з'єднання збирається в одному віртуальному контейнері, щоб трансформуватися в єдиний початковий запит, а потім відправляється користувачеві. Одне з'єднання може одночасно обробляти до 1024 запитів кінцевого користувача. Для зменшення навантаження на оперативну пам'ять веб-сервер використовує виділений сегмент пам'яті, який називається пул (pool). Він динамічний і розширюється зі збільшенням довжини запиту. У Nginx вбудовані механізми захисту. Інформація передається шифрованим каналом через протоколи SSL/TLS.

Найчастіше практика, коли Nginx працює у зв'язці з Apache. Якщо запит приходить видачу статичного матеріалу, його обробляє Nginx. Якщо користувачеві знадобився динамічний контент (відео, графіка), то тут підключається Apache. Веб-сервер Nginx ідеально підходить для сайтів, на яких міститься в основному статичний контент. Він також здатний виступити як редирект для поштових сервісів або ролі проксі-сервера. Простота та гнучкість налаштування дозволяє масштабувати продукт без особливих зусиль.

**Хід роботи**

**Встановлення і налаштування SHH**

Налаштовуємо SSH

Перевірка наявності або встановлення пакету **OpenSSH**

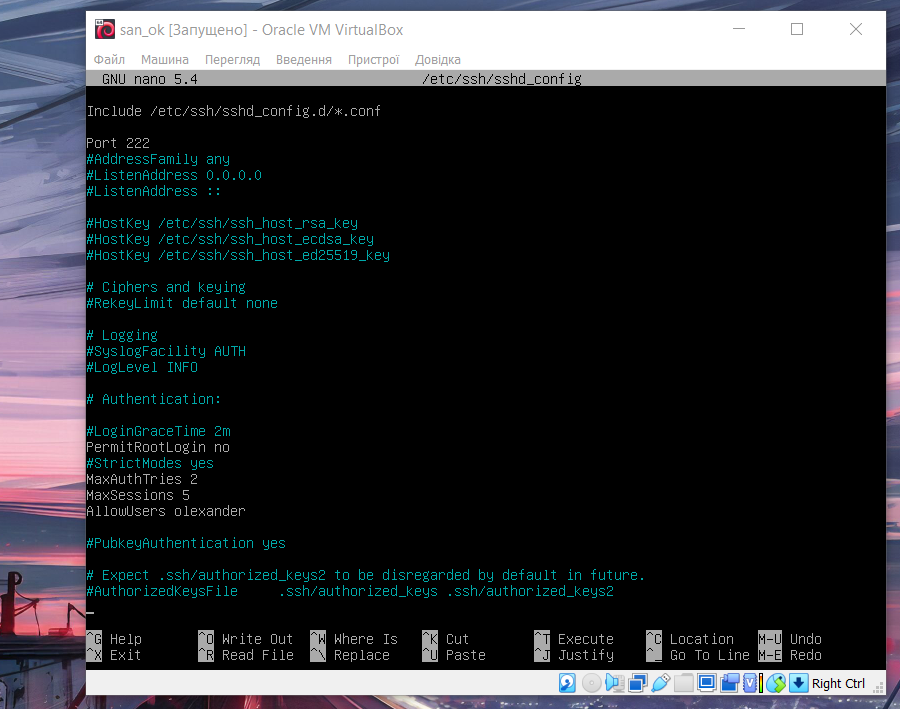


Якщо пакет відсутній, встановлюємо його, використовуючи команду:



Змінюємо налаштування SSH на необхідні нам за допомогою наступного файлу:

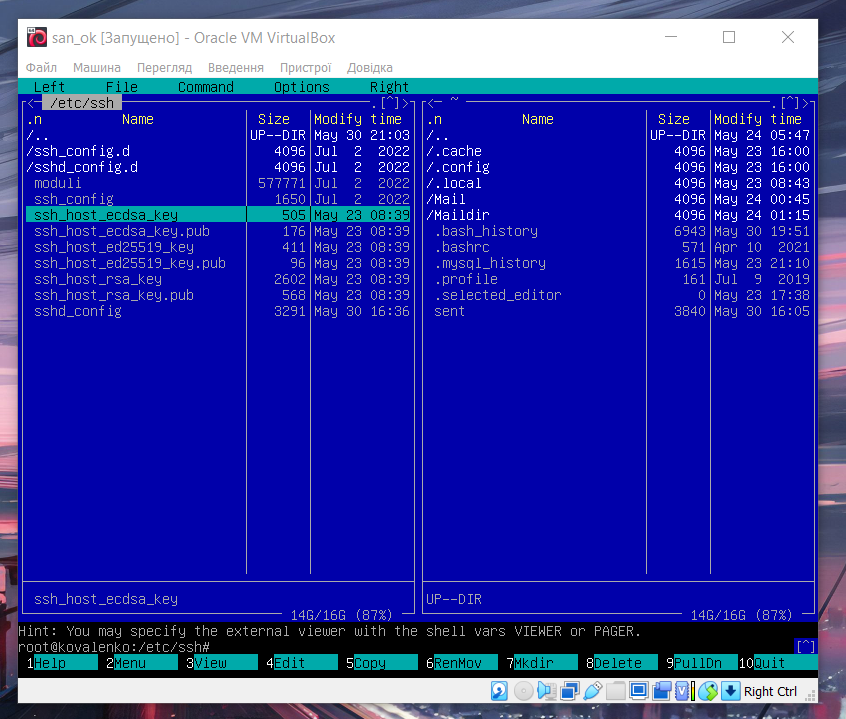


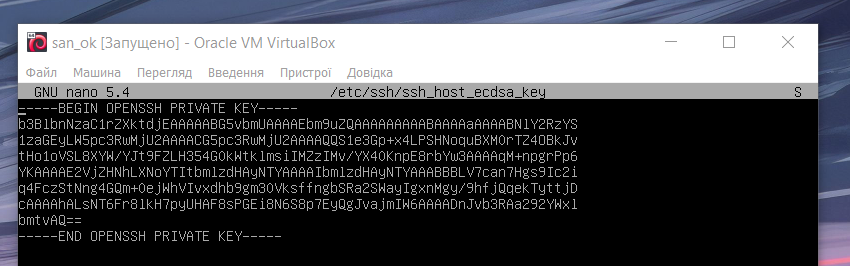


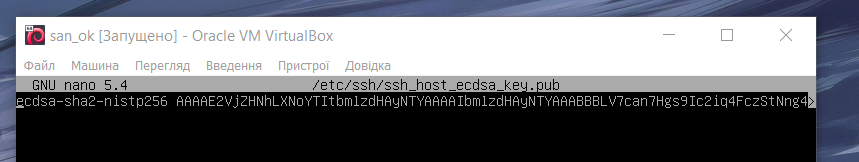
Перезавантажуємо SSH-сервер:

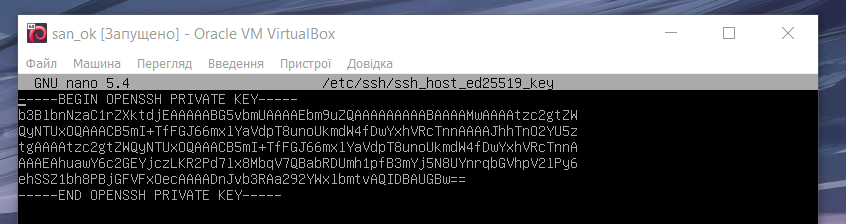
 або 

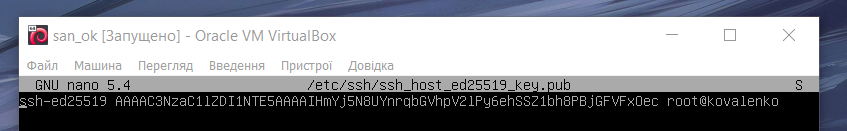
Перевіряємо наявність ключів шифрування:

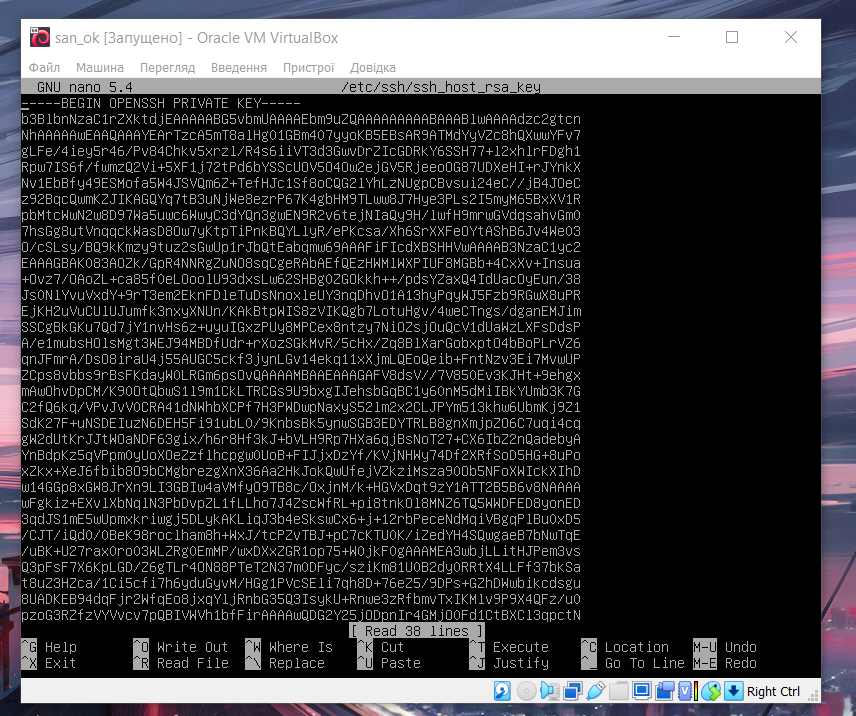


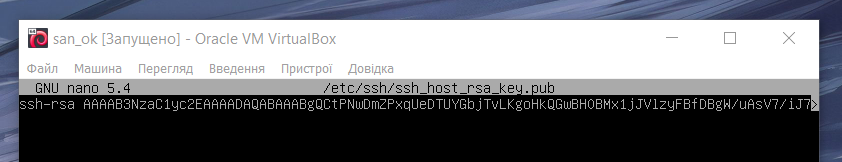




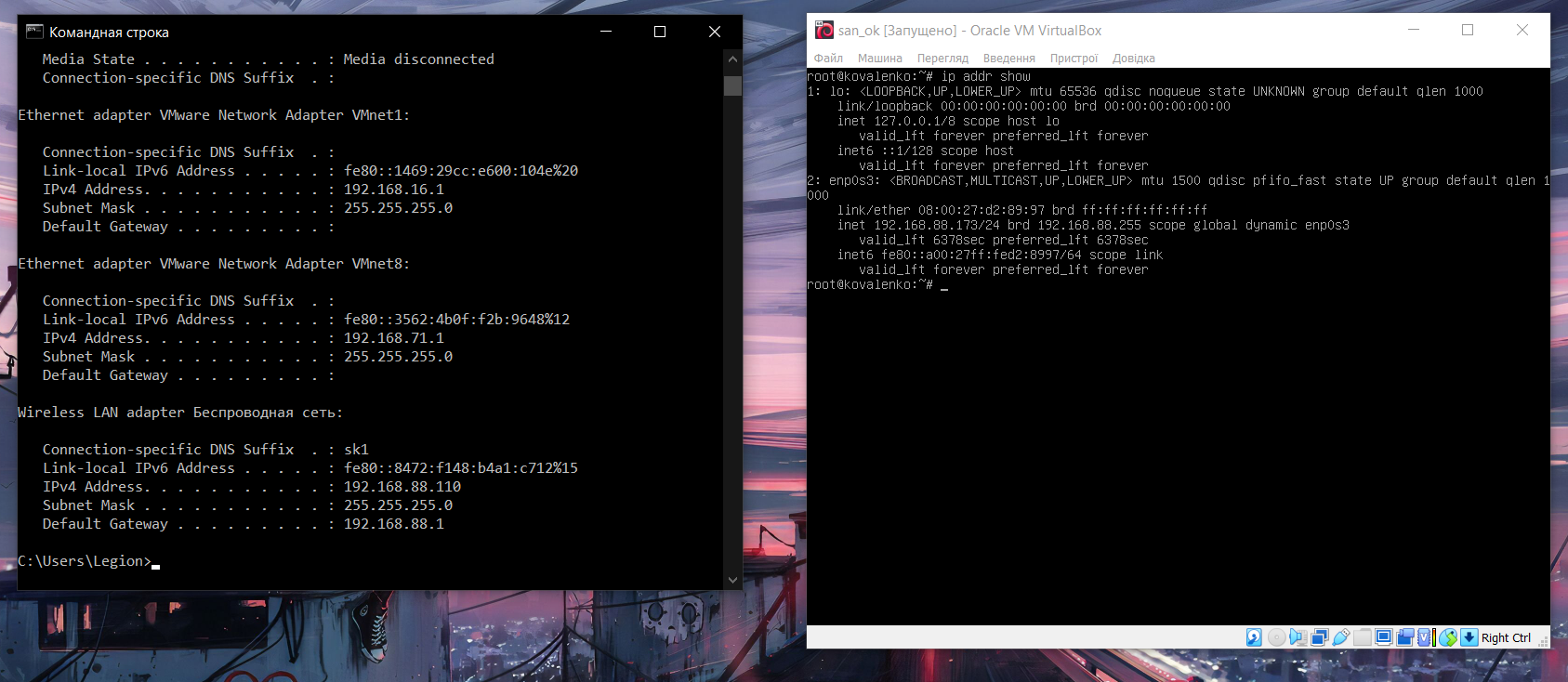




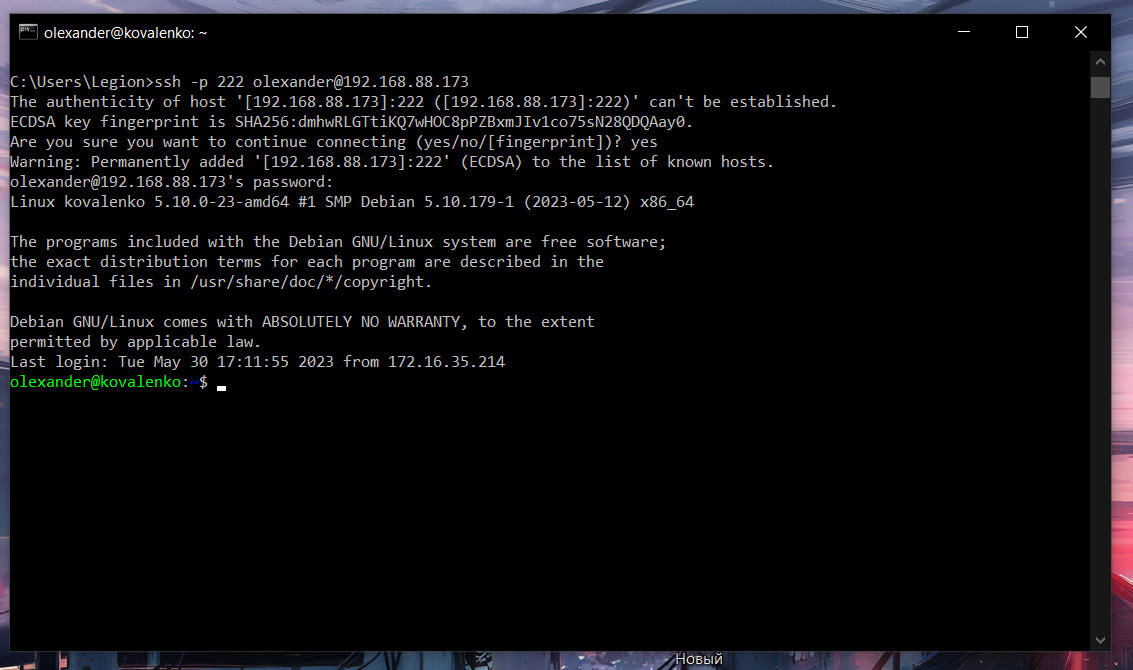




Переглядаємо поточні IP-адреси на віртуальній машині та на нашому комп’ютері:



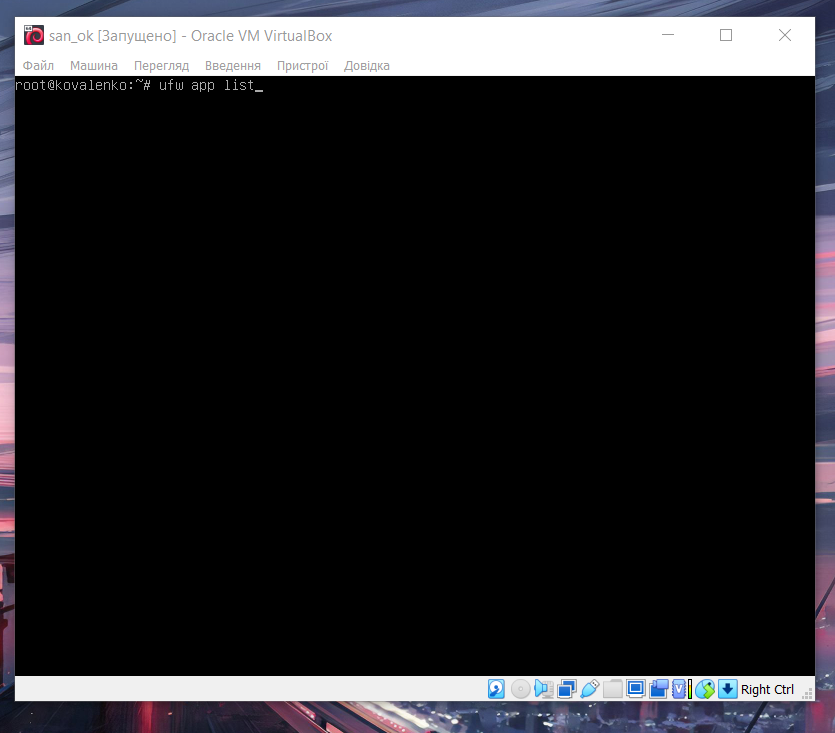
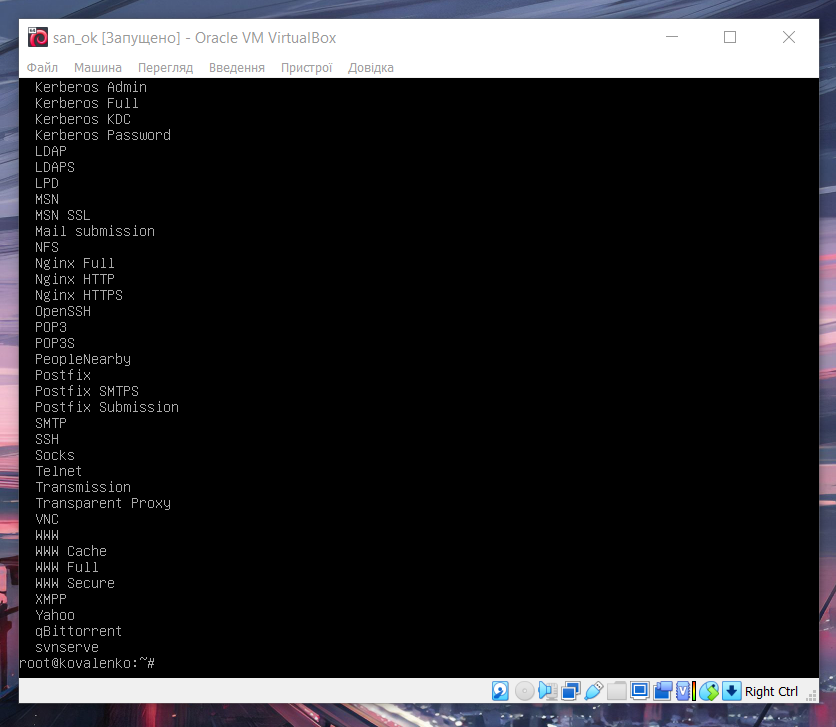
Перевірка чи працює SSH:



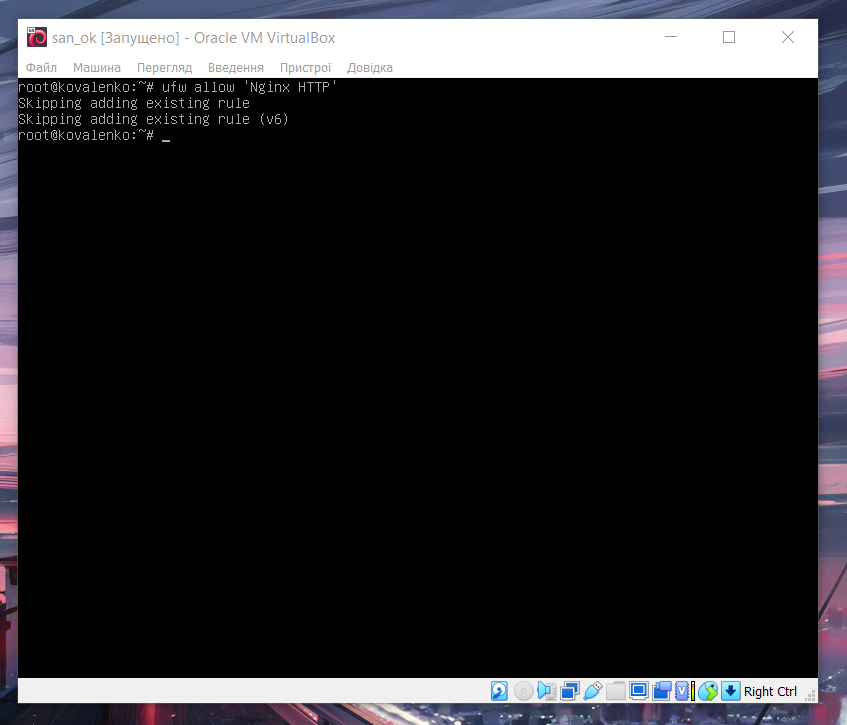
Отже, все налаштовано правильно. Тепер маємо можливість віддаленого підключення за допомогою стороннього комп’ютера.

**Встановлення і налаштування Nginx**

Переглядаємо доступні протоколи:

Обираємо і активуємо потрібний протокол:



Встановлення та конфігурування доменних імен:

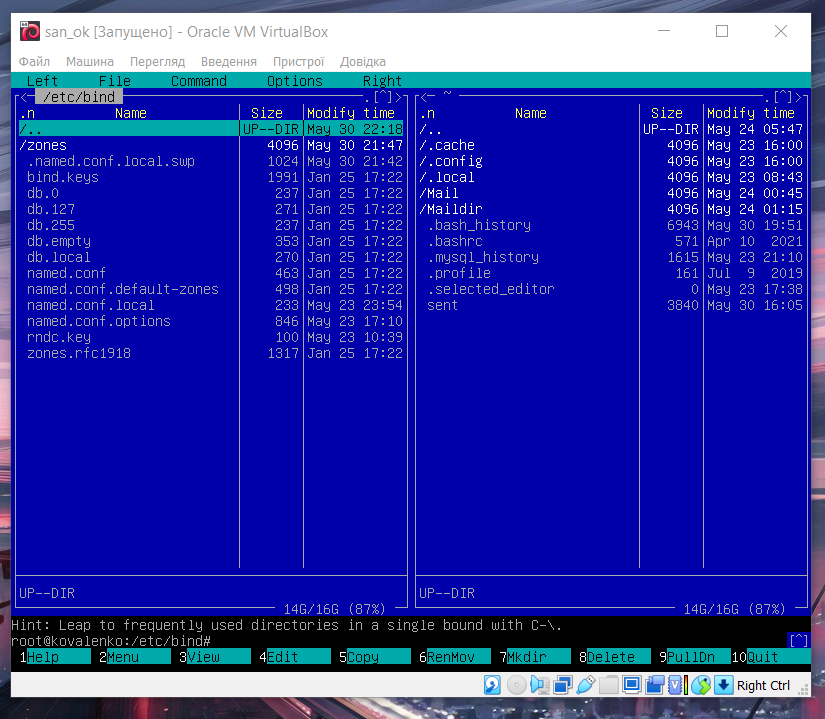
* встановити *bind9* та *dnsutils*;
* запустити bind9;
* додати адресу *dns*-сервера *google* (8.8.8.8) в якості форвардера для нашого сервера, за допомогою файлу */etc/bind/named.conf.options*;
* запевнитись, що конфігураційний файл */etc/bind/named.conf* включає в себе налаштування з файлів:
  + /etc/bind/named.conf.options
  + /etc/bind/named.conf.local
  + /etc/bind/named.conf.default-zones

Створимо пряму та зворотні зони, та сконфігуруємо їх.

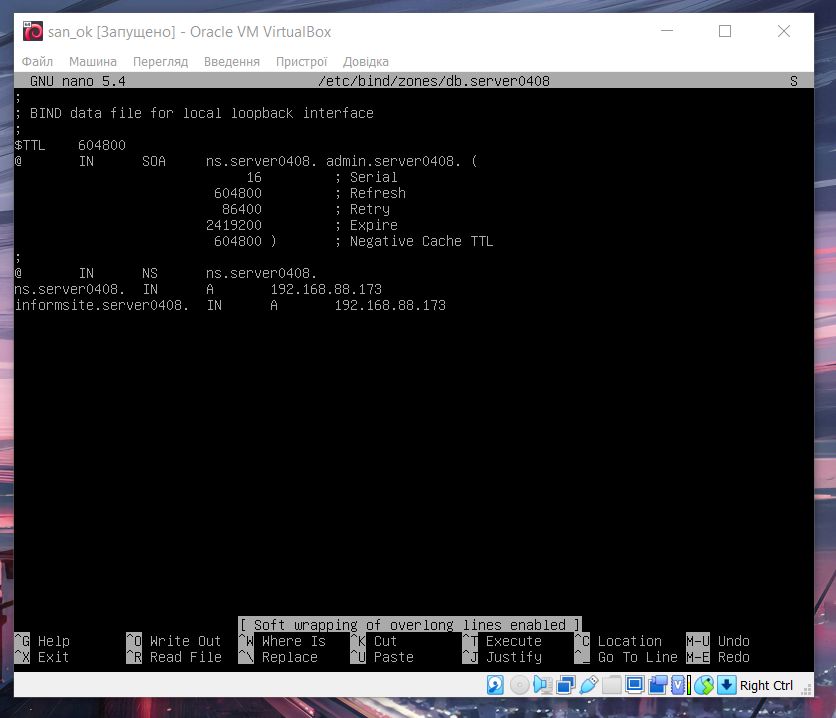
Для конфігурації виконайте наступне:

* cтворіть папку */etc/bind/zones*/ (в ній будуть зберігатись файли наших зон);
* створіть файл прямої зони
* створіть файл зворотної зони
* додайте опис прямої та зворотньої зони до файлу

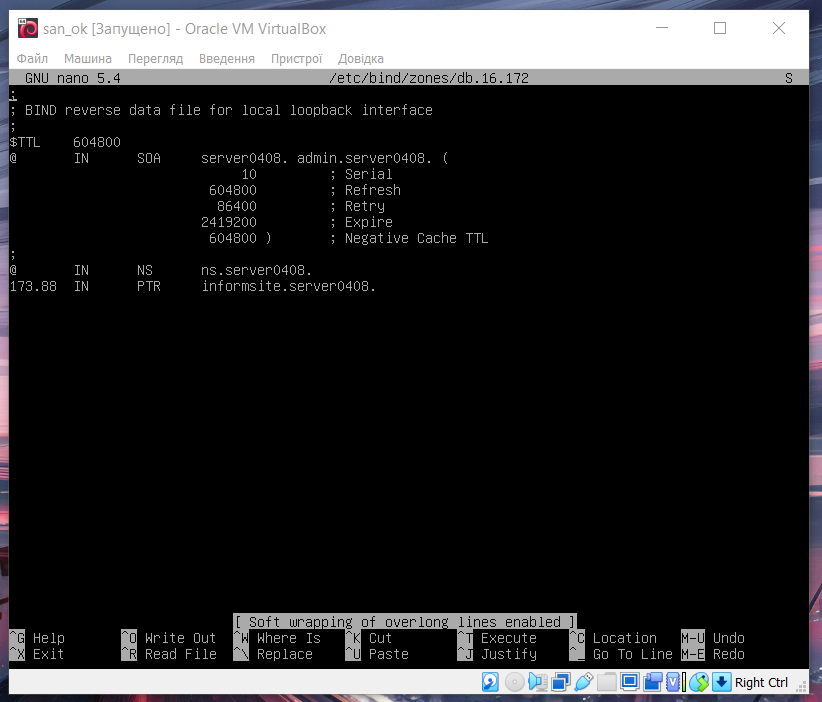
Переходимо до папки з файлами:



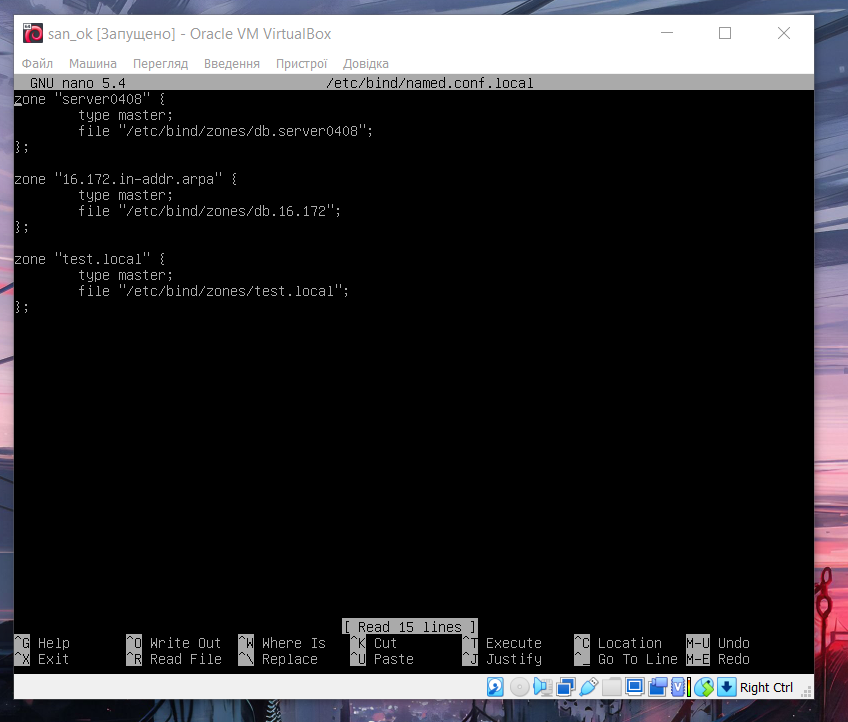
Файл прямої зони:



Файл зворотньої зони:

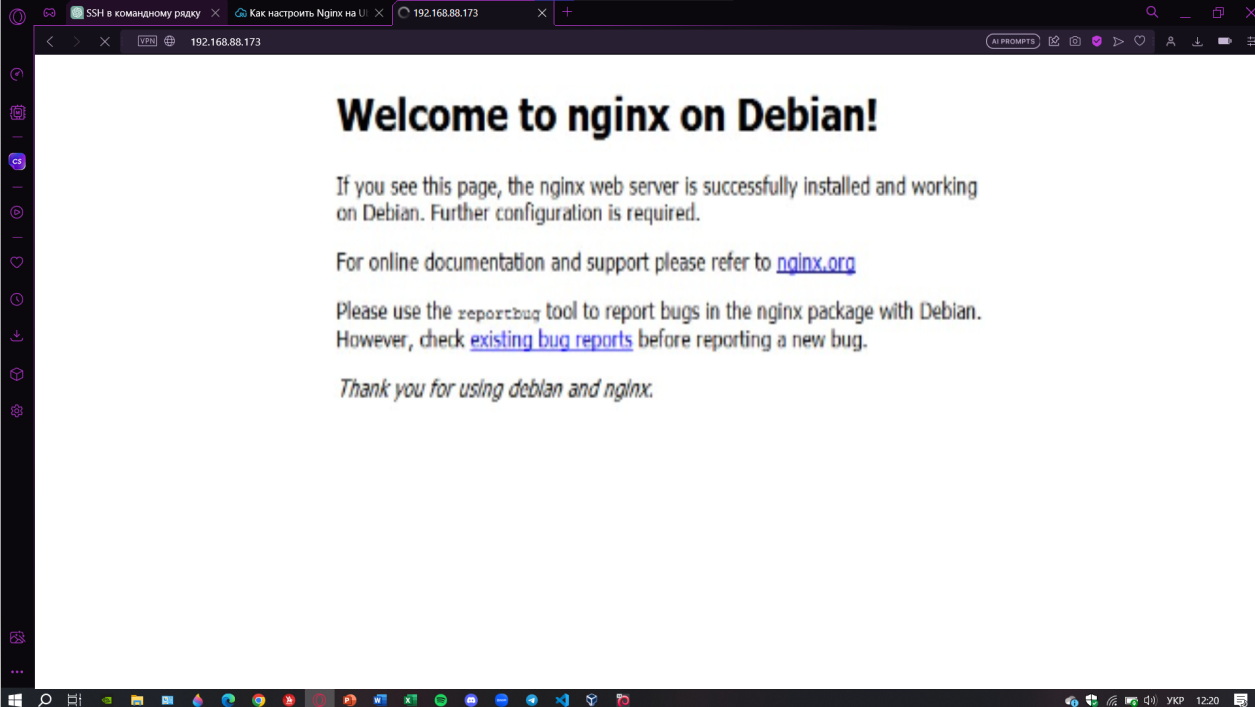


Опис прямої та зворотньої зон:



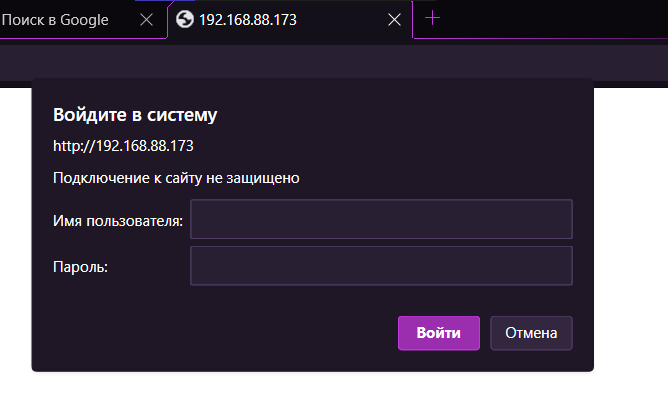
Залишається лише активувати сайт та перезавантажити сервіс

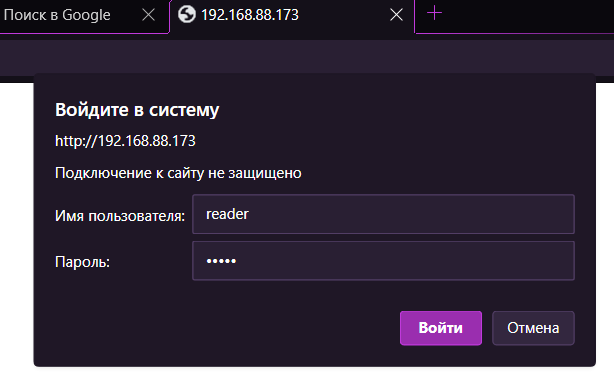
Маємо сайт:

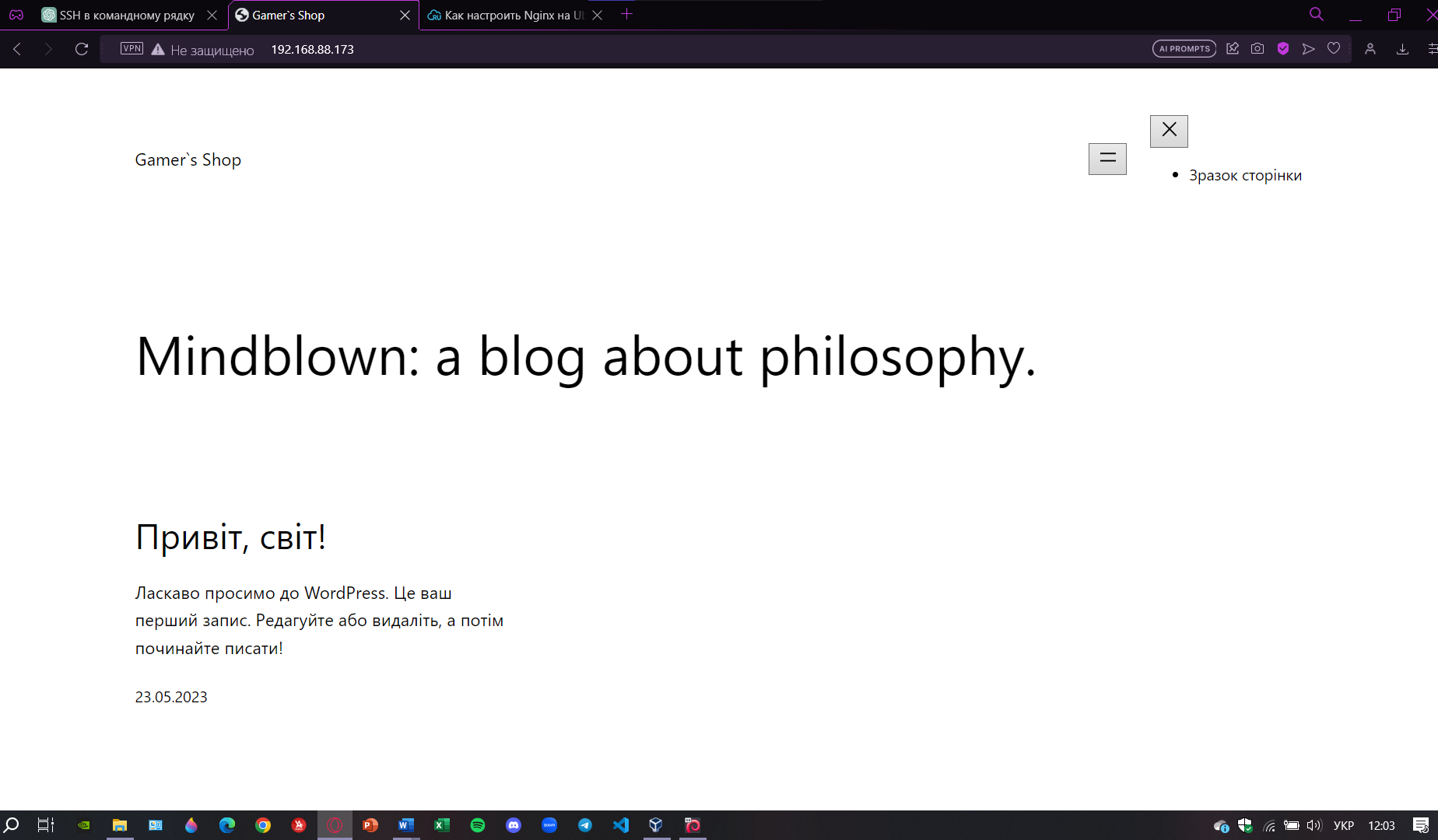


Крім цього, раніше було встановлено Apache-сервер і на ньому створено безпосередньо сайт:

Сайт запрошує введення логіну та паролю дозволеного користувача.







**Висновок**

SSH (Secure Shell) є безпечним протоколом для віддаленого доступу і керування комп'ютерами та серверами. Він забезпечує шифрування трафіку та механізми аутентифікації, що забезпечують конфіденційність і цілісність даних, а також запобігають несанкціонованому доступу. Переваги SSH включають можливість віддаленого доступу, передачу файлів, тунелювання портів та інші функції, що роблять його незамінним для системних адміністраторів, розробників та користувачів.

Nginx є швидким і легким веб-сервером, який широко використовується для обробки запитів HTTP і HTTPS, а також як проксі-сервер і балансувальник навантаження. Він має високу продуктивність, здатність обробляти великий обсяг запитів і відмінну масштабованість. Nginx також може бути налаштований як веб-сервер для статичних файлів і веб-сайтів, а також як зворотний проксі для перенаправлення запитів до різних серверів.

Обидва SSH і Nginx є потужними інструментами для розробників, системних адміністраторів і користувачів, які працюють з веб-серверами і віддаленими серверами. SSH надає безпечний доступ до віддалених систем і дозволяє виконувати різноманітні завдання. Nginx, з свого боку, забезпечує швидку та ефективну обробку запитів і може бути використаний для розгортання веб-сайтів та керування трафіком. Разом вони допомагають забезпечувати безпеку, продуктивність та надійність веб-проектів.