**ІНСТИТУТ СПЕЦІАЛЬНОГО ЗВ’ЯЗКУ ТА ЗАХИСТУ ІНФОРМАЦІЇ НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**

**ІМЕНІ ІГОРЯ СІКОРСЬКОГО”**

**КАФЕДРА № 1**

**ПРАКТИЧНА РОБОТА №3.3**

з навчальної дисципліни ‟Засоби і комплекси криптографічного захисту інформації”

**Тема: “Secure Shell”**

Виконав: курсант навчальної групи С-04

молодший сержант Павло ПАВЛЕНКО

*(підпис)*

Перевірив: капітан Володимир КУБРАК

*(підпис)*

КИЇВ – 2023

**ЗМІСТ**

[ВСТУП 3](#_Toc138415306)

[1. ЗАГАЛЬНІ ВІДОМОСТІ 5](#_Toc138415307)

[1.1. SECURE SHELL (SSH) 5](#_Toc138415308)

[1.2. SSH – KEY GENERATION 6](#_Toc138415309)

[1.3. ВАРІАЦІЇ ШИФРУВАННЯ КЛЮЧІВ 7](#_Toc138415310)

[1.4. ПЕРЕБРОС ПОРТІВ 8](#_Toc138415311)

[2. ПРАКТИЧНА РЕАЛІЗАЦІЯ 10](#_Toc138415312)

[2.1. НАЛАШТУВАННЯ SSH 10](#_Toc138415313)

[2.2 ГЕНЕРАЦІЯ КЛЮЧІВ 11](#_Toc138415314)

[3. ГЕНЕРАЦІЯ КЛЮЧІВ (РІЗНІ ВИДИ ШИФРУВАННЯ) 13](#_Toc138415315)

[3.1 RSA 13](#_Toc138415316)

[3.2. СПЕЦИФІЧНА ДОВЖИНА 13](#_Toc138415317)

[3.3. ECDSA 14](#_Toc138415318)

[3.4. Ed25519 14](#_Toc138415319)

[3.4. ПАРАНОЇДАЛЬНИЙ КЛЮЧ 15](#_Toc138415320)

[4. СТВОРЕННЯ ВЕБ-САЙТУ 16](#_Toc138415321)

[4.1. NGINX 16](#_Toc138415322)

[4.2. ПЕРЕНАПРАВЛЕННЯ ПОРТІВ ТА КОНФІГУРАЦІЯ 17](#_Toc138415323)

[ВИСНОВКИ 19](#_Toc138415324)

# ВСТУП

SSH (Secure Shell) - це протокол, що забезпечує безпечний зв'язок між комп'ютерами через незахищені мережі. Він дозволяє нам здійснювати віддалений доступ до інших комп'ютерів, передавати файли та виконувати різноманітні команди за допомогою зашифрованого з'єднання.

У нашому практичному занятті ми розглянемо різні аспекти SSH та навчимося працювати з ним на прикладі віртуальної машини на базі Ubuntu та локальної машини на Windows.

Спочатку ми дізнаємось загальні відомості про SSH, його призначення та переваги. Розберемося, як встановити SSH-сервер та клієнт, які дозволять нам створити з'єднання між комп'ютерами.

Потім ми розглянемо процес генерації ключів SSH. Ключі використовуються для ідентифікації та аутентифікації користувачів. Ми дослідимо різні варіації шифрування ключів та розберемося, як правильно згенерувати та зберегти ключі SSH.

Наступним кроком буде налаштування SSH-з'єднання між віртуальною машиною на Ubuntu та локальною машиною на Windows. Ми дізнаємось, як з'єднатися з віртуальною машиною за допомогою SSH, використовуючи IP-адресу та правильні облікові дані.

Після успішного підключення ми згенеруємо ключ для SSH, що дозволить нам безпечно авторизуватись на віртуальній машині. Ми перевіримо працездатність нашого з'єднання та впевнимося, що все налаштовано правильно.

Нарешті, ми створимо невеликий веб-сайт "Hello World!" на нашій віртуальній машині, щоб перевірити, як SSH дозволяє нам виконувати різні дії на віддаленому сервері.

На завершення нашого заняття ми також спробуємо налаштувати переброс портів, щоб забезпечити доступ до веб-сервера на віртуальній машині через порт 80 та локальну машину.

Під час нашої практики ми будемо крок за кроком виконувати ці дії, розглядаючи кожен етап детально та надаючи практичні приклади. Без сумніву, це пізнавальне заняття, яке допоможе нам краще розібратись зі світом SSH та навчитись використовувати його в реальних ситуаціях.

# 1. ЗАГАЛЬНІ ВІДОМОСТІ

## 1.1. SECURE SHELL (SSH)

**Secure Shell (SSH)** - це протокол, який дозволяє безпечно з'єднуватися і керувати віддаленим комп'ютером або сервером через інтернет. Основна його ідея полягає в тому, щоб передавати дані між комп'ютерами зашифровано, щоб інші люди не могли перехопити цю інформацію.

**SSH** **забезпечує** дві головні речі:

1. аутентифікацію (перевірку вашої ідентичності);
2. шифрування (захист даних від несанкціонованого доступу).

**Переваги SSH:**

* Безпека: SSH забезпечує шифрування з'єднання між клієнтом та сервером, що робить його дуже безпечним для передачі конфіденційної інформації через незахищені мережі.
* Аутентифікація: SSH використовує ключі або паролі для аутентифікації клієнтів, що робить процес підключення більш безпечним та захищеним від перехоплення.
* Віддалений доступ: SSH дозволяє вам віддалено управляти комп'ютером або сервером через командний рядок, що забезпечує зручний доступ до системи, навіть якщо ви знаходитесь в іншому місці.
* Переносимість: SSH є стандартним протоколом та підтримується на багатьох операційних системах, що дозволяє легко встановлювати з'єднання між різними платформами.

**Недоліки SSH:**

* Складність налаштування: Правильна настройка SSH може вимагати деяких технічних знань та розуміння мережевих налаштувань, що може бути складним для початківців.
* Залежність від мережі: SSH вимагає працездатної мережі для здійснення з'єднання. Якщо мережа нестабільна або відсутня, це може призвести до проблем з підключенням.
* Наявність вразливостей: Як і будь-який інший протокол, SSH може мати вразливості, які можуть бути використані зловмисниками для атак на систему. Важливо регулярно оновлювати SSH та вживати заходів безпеки для забезпечення найвищого рівня захисту.Коли ви підключаєтесь до віддаленого комп'ютера за допомогою SSH, ви повинні надати свої користувацькі дані (логін та пароль або ключ), щоб довести, що ви маєте право доступу до цього комп'ютера.

SSH є дуже корисним інструментом для адміністрування серверів або виконання роботи на віддалених комп'ютерах, особливо в ситуаціях, коли ви не знаходитесь фізично поряд з ними. Ви можете виконувати команди в терміналі, передавати файли та керувати системними налаштуваннями, все це через зашифроване з'єднання.

Для використання SSH вам знадобиться SSH-клієнт на вашому комп'ютері, який буде встановлювати з'єднання з віддаленим комп'ютером, і SSH-сервер на віддаленому комп'ютері, до якого ви хочете підключитися.

## 1.2. SSH – KEY GENERATION

Процес генерації ключів SSH (ssh-keygen) використовується для створення пари ключів - приватного і публічного, які використовуються для забезпечення безпечного з'єднання SSH і аутентифікації на віддаленому сервері.

Ось простий **опис процесу генерації ключів SSH:**

1. **Виклик команди ssh-keygen:** Ви запускаєте команду ssh-keygen на своєму комп'ютері. Ця команда генерує пару ключів - приватний і публічний - за замовчуванням.
2. **Вибір розташування та назви файлів ключів:** Вас можуть попросити вказати розташування та назву файлів для збереження ключів. Зазвичай приватний ключ зберігається в файлі з розширенням .ssh/id\_rsa, а публічний ключ - у файлі з розширенням .ssh/id\_rsa.pub. Але ви можете вибрати будь-яке інше розташування та назву за бажанням.
3. **Генерація ключів:** Команда ssh-keygen генерує випадкову пару ключів - приватний і публічний. Приватний ключ зберігається на вашому комп'ютері, а публічний ключ можна передати на віддалений сервер.
4. **Встановлення фрази-пароля (необов'язково):** Вас можуть попросити ввести фразу-пароль для захисту приватного ключа. Це додатковий шар безпеки, який зашифрує ваш приватний ключ за допомогою введеної фрази-пароля. Це необов'язковий крок, але рекомендується для забезпечення безпеки ключів.
5. **Завершення процесу:** Після завершення генерації ключів ви отримаєте повідомлення про успішне створення ключів.

## 1.3. ВАРІАЦІЇ ШИФРУВАННЯ КЛЮЧІВ

Тепер щодо різних варіантів шифрування ключів SSH. При генерації ключів SSH можна вибрати різні алгоритми шифрування ключів. Основні варіанти включають:

1. **RSA:** Це один з найпоширеніших алгоритмів шифрування використовується для генерації ключів SSH. Він надійний і широко підтримується різними SSH-серверами і клієнтами.
2. **DSA:** Цей алгоритм шифрування був популярним раніше, але зараз рекомендується використовувати RSA замість нього.
3. **ECDSA:** Це алгоритм шифрування, який використовує еліптичні криві. Він забезпечує сильну безпеку при меньшому розмірі ключа порівняно з RSA.
4. **Ed25519:** Це новітній алгоритм шифрування, який надає високу безпеку і продуктивність. Він стає все більш популярним і рекомендується для нових встановлень SSH ключів.

Кожен алгоритм має свої переваги і може бути використаний для генерації ключів SSH. Однак, рекомендовано використовувати більш сучасні алгоритми, такі як RSA, ECDSA або Ed25519, оскільки вони надають кращу безпеку та продуктивність.

## 1.4. ПЕРЕБРОС ПОРТІВ

**Переброс портів (port forwarding)** – це процес пересилання мережевого трафіку з одного порту на одному пристрої до іншого порту на іншому пристрої. Це корисний механізм, який дозволяє направляти запити до конкретних послуг або додатків, які працюють на віддаленому сервері чи внутрішній мережі.

Існує два типи перебросу портів: **локальний** (Local Port Forwarding) і **віддалений** (Remote Port Forwarding).

1. **Локальний переброс портів (Local Port Forwarding):**

Локальний переброс портів використовується для пересилання мережевого трафіку з локального комп'ютера до віддаленого сервера. Це дозволяє отримувати доступ до служб або додатків, які запущені на віддаленому сервері через локальний комп'ютер.

Приклад команди для локального перебросу портів через SSH (рис.1.4.1):



Рис.1.4.1. – Локальний переброс портів

В цій команді local\_port - це порт на локальному комп'ютері, remote\_address - адреса віддаленого сервера, remote\_port - порт на віддаленому сервері.

1. **Віддалений переброс портів (Remote Port Forwarding):**

Віддалений переброс портів використовується для пересилання мережевого трафіку з віддаленого сервера до локального комп'ютера. Це дозволяє отримувати доступ до служб або додатків, які запущені на локальному комп'ютері через віддалений сервер.

Приклад команди для віддаленого перебросу портів через SSH (рис.1.4.2.):



Рис.1.4.2. – Віддалений переброс портів

В цій команді remote\_port - це порт на віддаленому сервері, local\_address - IP-адреса локального комп'ютера (зазвичай localhost або 127.0.0.1), local\_port - порт на локальному комп'ютері.

Переброс портів є корисним інструментом для надання зовнішнього доступу до різноманітних служб і додатків через захищене SSH-з'єднання. Він використовується для різних сценаріїв, таких як віддалений доступ до веб-сайтів, віддалене керування базами даних, віддалений доступ до внутрішніх мереж і багато іншого.

# 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ

## 2.1. НАЛАШТУВАННЯ SSH

Перевірка наявності SSH на віртуальній машині (рис.2.1.1.):

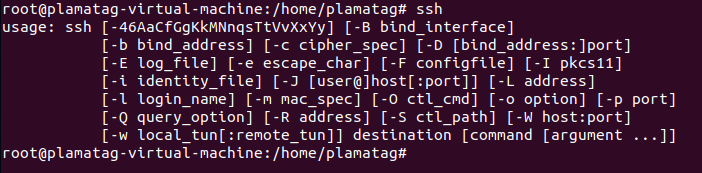


Рис.2.1.1. – Перевірка наявності SSH (віртуальна машина)

Перевірка статусу SSH на віртуальній машині (рис.2.1.2.):

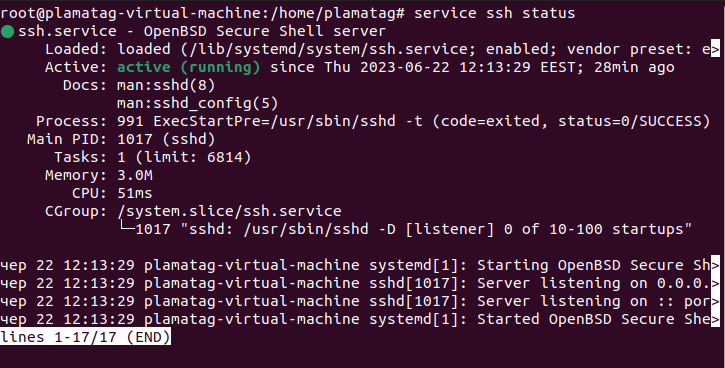


Рис.2.1.2. – Перевірка статусу SSH (віртуальна машина)

Перевірка наявності SSH на локальній машині (рис.2.3.):

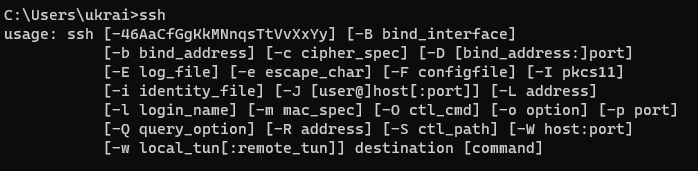


Рис.2.1.3. – Перевірка наявності SSH (локальна машина)

## 2.2 ГЕНЕРАЦІЯ КЛЮЧІВ

Згенеруємо ключ SSH (рис.2.2.1):

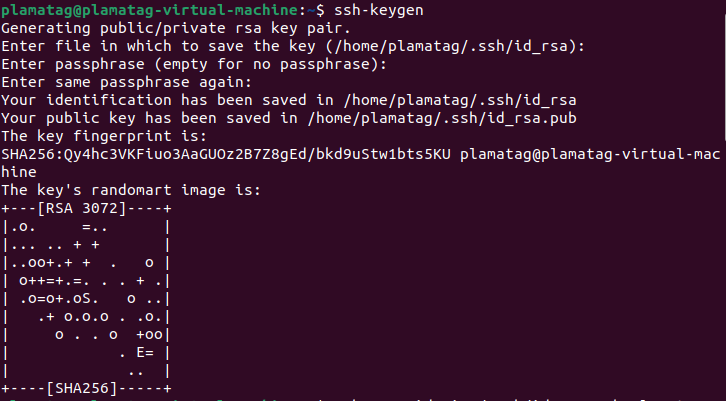


Рис.2.2.1. – Генерація ключа SSH

Перевірка наявності публічного ключа SSH (рис.2.2.2.):

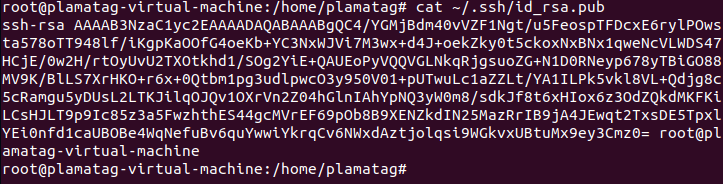


Рис.2.2.2. – Наявність публічного ключа SSH

Дізнаємось IP-адресу сервера (рис.2.2.3):

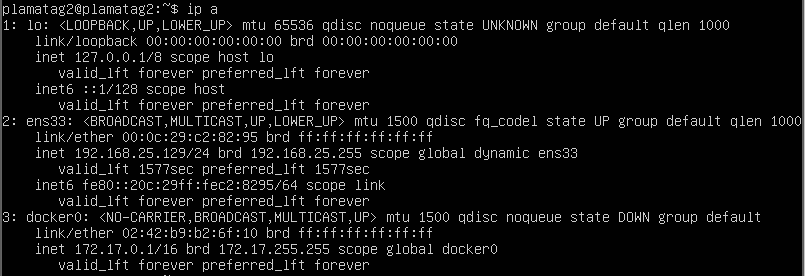


Рис.2.2.3 – IP-адреса сервера

Копіювання публічного ключа на сервер (рис.2.2.4.):

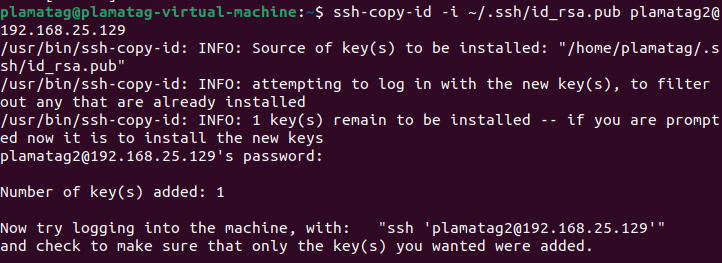


Рис.2.2.4 – Копіювання публічного ключа на сервер

Перевірка працездатності SSH та ключа, тобто при підключенні не треба вводити пароль (рис.2.2.5.):

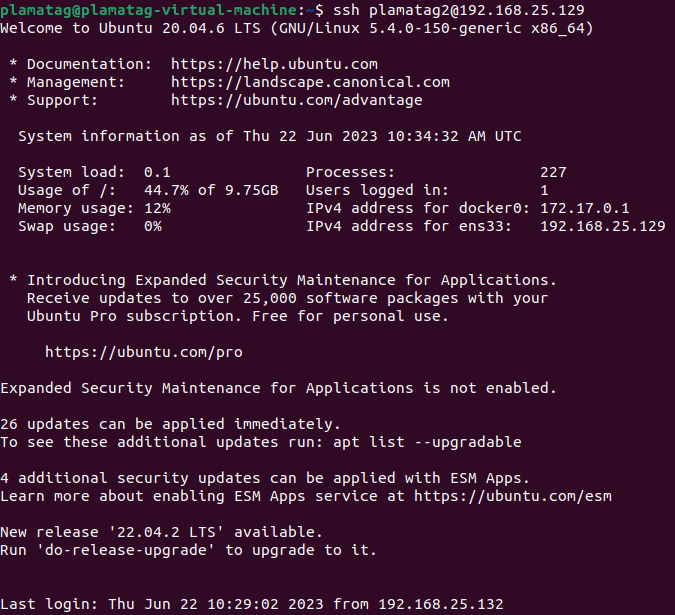


Рис.2.2.5. – Перевірка працездатності SSH та ключа

# 3. ГЕНЕРАЦІЯ КЛЮЧІВ (РІЗНІ ВИДИ ШИФРУВАННЯ)

## 3.1 RSA

Створення ключа за замовчуванням (RSA) (рис.3.1.):

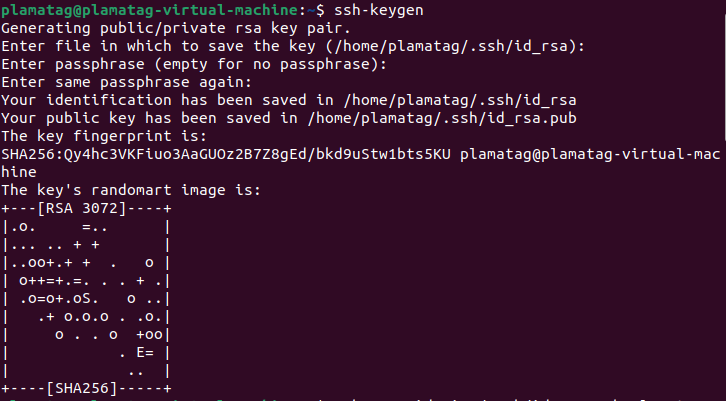


Рис.3.1. – RSA

## 3.2. СПЕЦИФІЧНА ДОВЖИНА

Створення ключа зі специфічною довжиною (рис.3.2.):

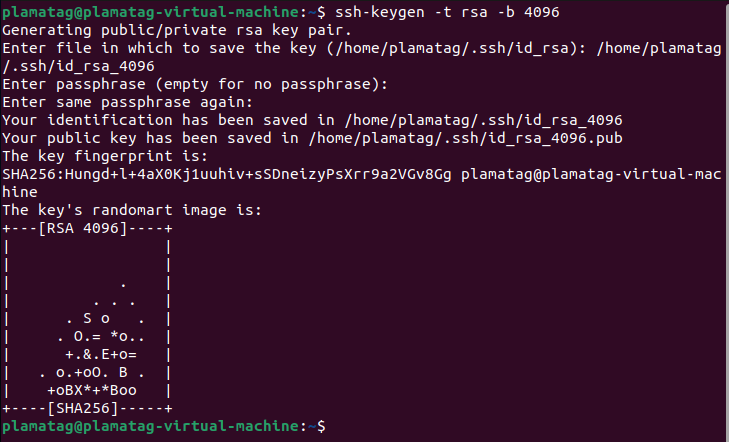


Рис.3.2. – Створення ключа зі специфічною довжиною

## 3.3. ECDSA

Створення ключа з алгоритмом ECDSA (рис.3.3.):

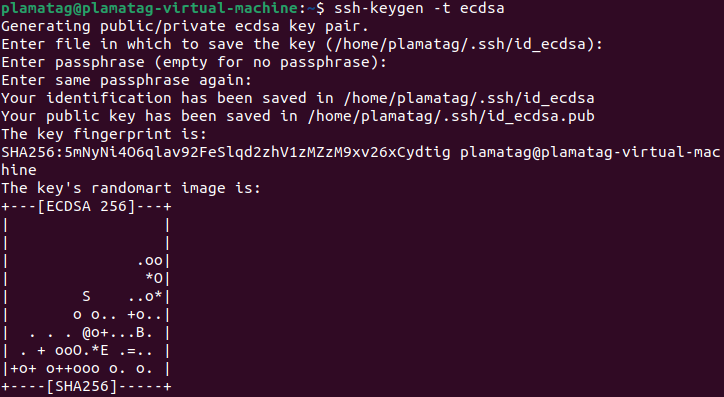


Рис.3.3. – ECDSA

## 3.4. Ed25519

Створення ключа з алгоритмом Ed25519 (рис.3.4.):

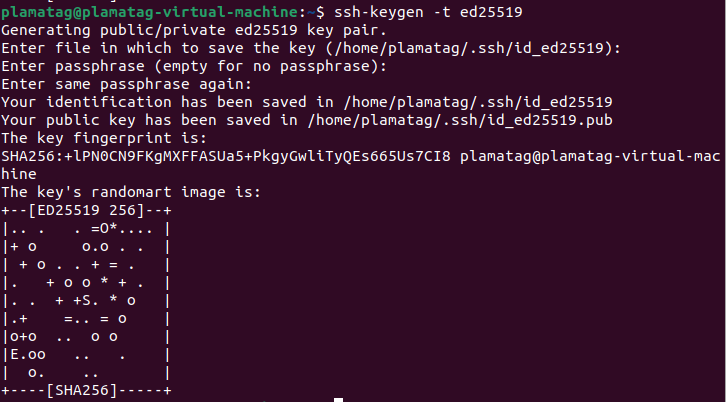


Рис.3.4. - Ed25519

## 3.4. ПАРАНОЇДАЛЬНИЙ КЛЮЧ

Створення параноїдального ключа (рис.3.4.1.):

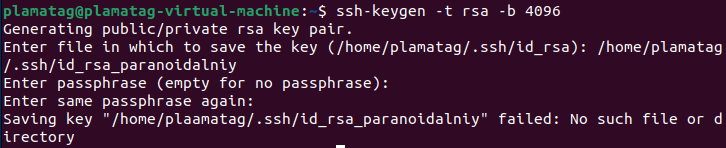


Рис.3.4.1. – Параноїдальний ключ

# 4. СТВОРЕННЯ ВЕБ-САЙТУ

## 4.1. NGINX

Під’єднаємось до серверу (рис.4.1.1.):

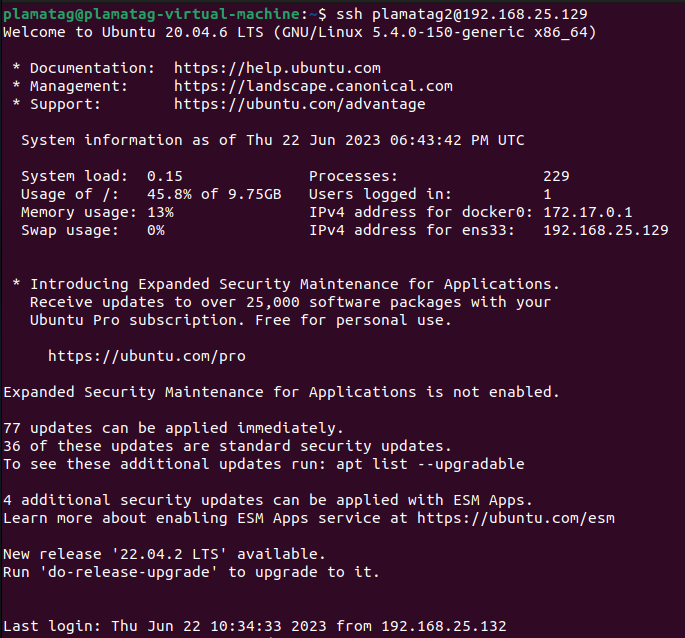


Рис.4.1.1. – Під’єднання до серверу

Встановимо веб-сервер nginx на сервері (рис.4.1.2.):



Рис.4.1.2. – Встановлення nginx

Створіть конфігураційний файл для вашого сайту (рис.4.3.):



Рис.4.1.3. – Конфігураційний файл

## 4.2. ПЕРЕНАПРАВЛЕННЯ ПОРТІВ ТА КОНФІГУРАЦІЯ

У цьому файлі ви можете визначити налаштування для вашого сайту, а також перенаправлення портів. Ось приклад базової конфігурації (рис.4.4.):

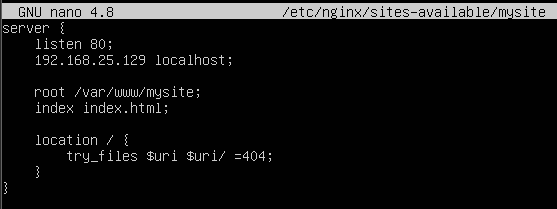


Рис.4.4. – Налаштування для вашого сайту

Активуйте конфігураційний файл. Створіть символічне посилання на ваш конфігураційний файл у каталозі (рис.4.2.5.):



Рис.4.2.5. – Активація конфігураційного файлу

Перезапустіть Nginx (рис.4.2.6.):



Рис.4.2.6. – Перезапуск Nginx

Тепер ви повинні налаштувати перенаправлення портів на вашій локальній машині. Відкрийте новий термінал або командний рядок на вашій локальній машині і введіть наступну команду (рис.4.2.7.):

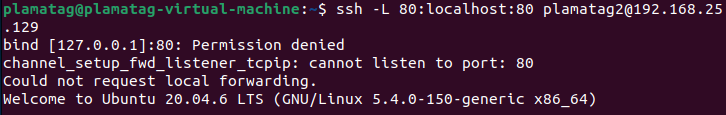


Рис.4.2.7. – Перенаправлення портів

Перевірте сайт: Відкрийте веб-браузер на вашій локальній машині і перейдіть за адресою http://localhost. Ви повинні бачити ваш сайт, який розміщений на віртуальному сервері через Nginx (рис.4.2.7.):

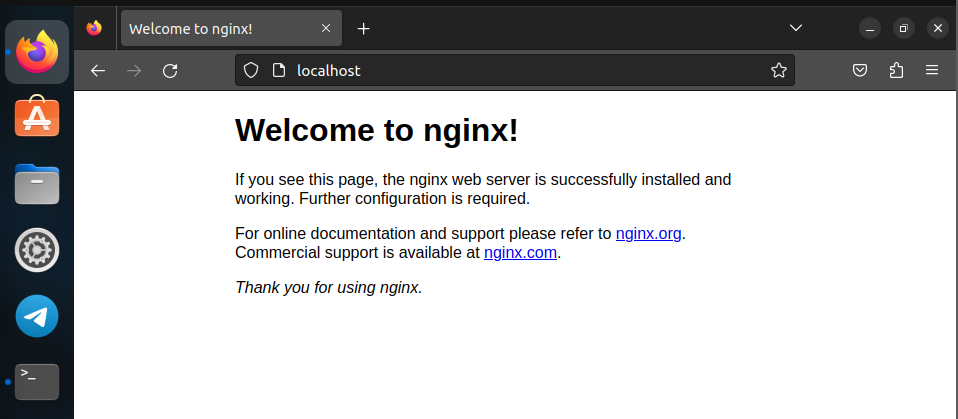


Рис.4.2.7. – Перевірка сайту

Останній крок, змінюємо вміст файлу на “Hello World!” (рис.4.2.8.):

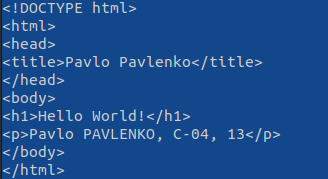


Рис.4.2.8. – Hello World!

Перевіряємо результат наших змін (рис.4.2.9.):

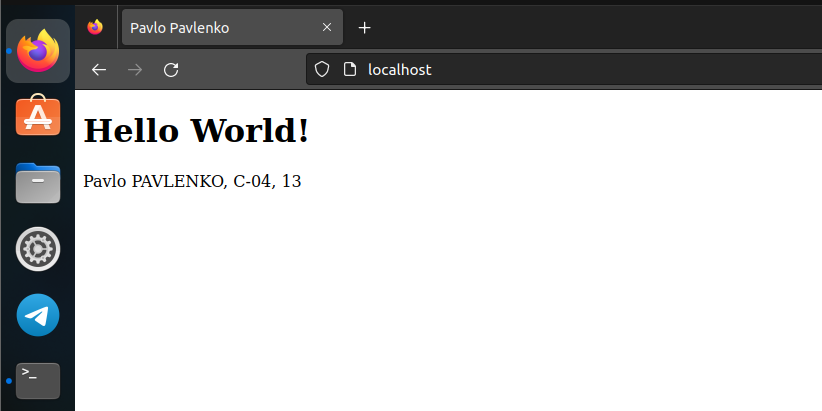


Рис.4.2.9. – Остаточний результат

# ВИСНОВКИ

Виходячи з виконання практичної роботи, зробимо наступні висновки:

1. SSH (Secure Shell) є безпечним протоколом для віддаленого доступу до серверів і забезпечує шифрування даних під час передачі.
2. Генерація ключів SSH є необхідним кроком для забезпечення аутентифікації та безпеки SSH-з'єднання.
3. Переброс портів (port forwarding) дозволяє пересилати мережевий трафік з одного порту на одному пристрої до іншого порту на іншому пристрої. Це корисний механізм для доступу до послуг і додатків, що працюють на віддалених серверах.
4. В налаштуваннях SSH можна використовувати різні види шифрування ключів, такі як RSA, ECDSA і Ed25519, залежно від потреб і рівня безпеки.
5. Nginx є популярним веб-сервером, який використовується для розгортання веб-сайтів і обробки HTTP-запитів.
6. Перенаправлення портів у Nginx може бути налаштоване для пересилання запитів з одного порту до іншого на локальній машині чи віддаленому сервері.

Усі ці аспекти – SSH, генерація ключів, переброс портів, веб-сервер Nginx – є важливими для налаштування безпечного віддаленого доступу та розгортання веб-сайтів. Вони надають можливість забезпечити захист і ефективність вашої мережевої інфраструктури.