**Simple Linux Network Scanner Web App**

* **Introduction**

This project merges Linux expertise with network scanning capabilities using nmap to develop a browser-accessible network scanner. The objective is to enable continuous monitoring of network status, updating every 10 minutes to facilitate timely intervention if necessary. PHP programming will be leveraged to convert network scan results into easily readable text files, ensuring accessibility via web browsers. This initiative underscores the practical application of Linux skills, effective account management, and SQL programming in creating a robust tool for network administrators.

* **Motivation**

This project is to provide a simple yet effective solution for network administrators to monitor and visualize their network's status in real-time. For individuals involved in IT administration or cybersecurity, undertaking such a project provides valuable hands-on experience with network scanning tools, web server configuration, and scripting. It enhances technical skills and familiarity with essential tools commonly used in these fields. The project is adaptable to various network environments and can be extended with additional functionalities as needed. This flexibility allows administrators to tailor the solution to specific organizational requirements or integrate with other security tools. Ultimately, the project aims to contribute to the operational efficiency of network management by streamlining the process of monitoring and maintaining network health. This can lead to quicker response times to network issues and improved overall network security posture.

* **What area of work?**

The project focuses on the integration of Linux system administration and network scanning capabilities to develop a web-based network scanner application. This involves utilizing tools like nmap for network exploration and security auditing within a local network environment.

* **Feasibility Study**

A feasibility study is a critical process undertaken to asses the practically of a proposed project, initiative, or business venture. It serves a comprehensive evaluation that examines various aspects such as technical, economic, legal, operational, and scheduling feasibility. The primary goal is to determine whether the project is viable and achievable within the defined constraints and objectives. Feasibility studies typically involve conducting research, analysing data, and forecasting outcomes to provide stakeholders with valuable insights to make informed decisions before committing resources.

* + Software required
* Linux - Required for running services like Apache, PHP, and managing cron jobs.
* Appache2 - To serve the PHP files and host the web interface.
* PHP - Used to process server-side scripts for dynamic content generation.
* Nmap - Network exploration tool and security/port scanner. It will scan your network and produce output files.
* Cron - This cron job runs every 10 minutes, scanning the network 192.168.1.0/24 with nmap and saving the output to /var/www/html/nmap.html.

* **Inputs**
* **Cron Job**: Configured to execute a nmap scan every 10 minutes on a defined network range.
* **nmap Command Output**: Saved to a designated file in a web-accessible directory.
* **Outputs**
  + **Formatted Network Status**
  + **Timestamp**
  + **Network Scan Results**: Includes detailed information such as host availability, open ports, and potentially vulnerable services discovered during the scan.

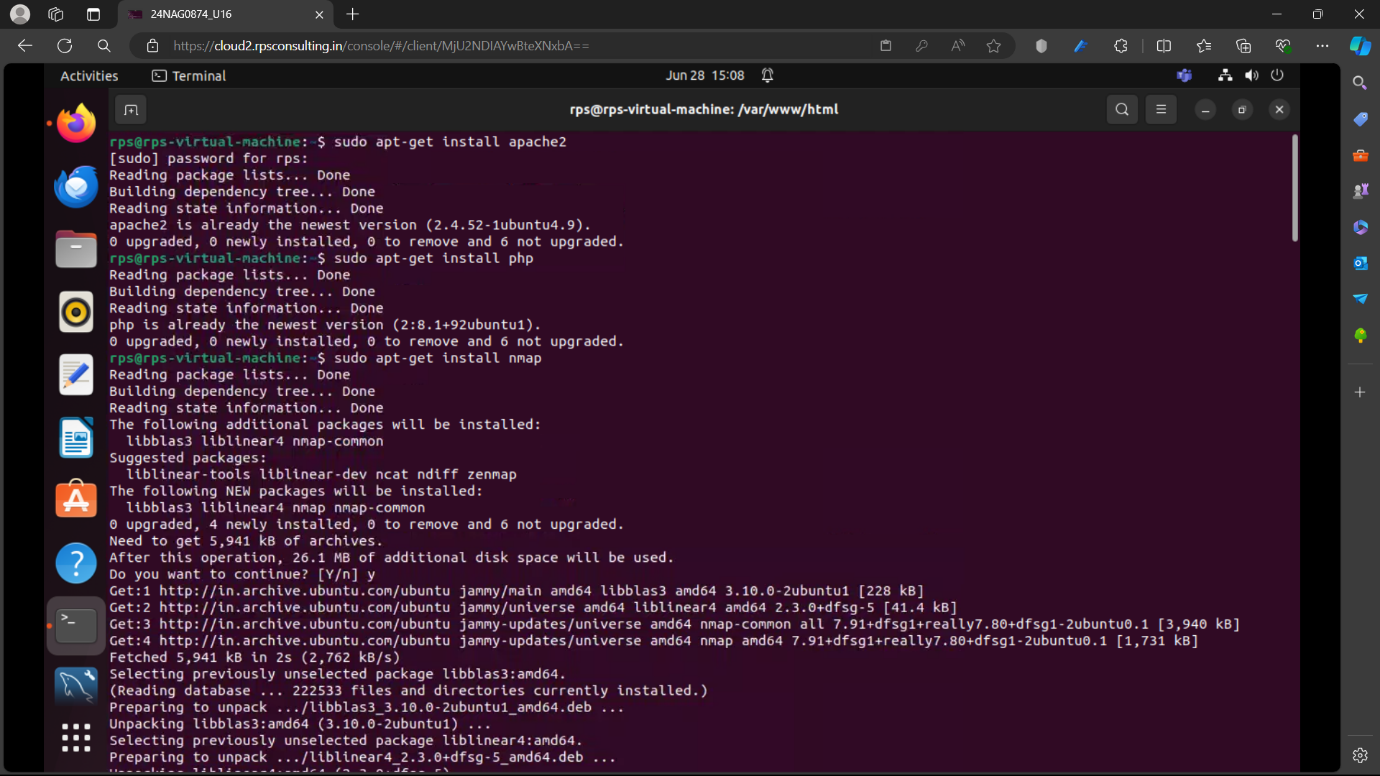
**Development and Testing**

* **Step 1: Install Required Software**

Ensure your Linux environment has Apache2, PHP, and nmap installed.

sudo apt-get update

sudo apt-get install apache2 php nmap

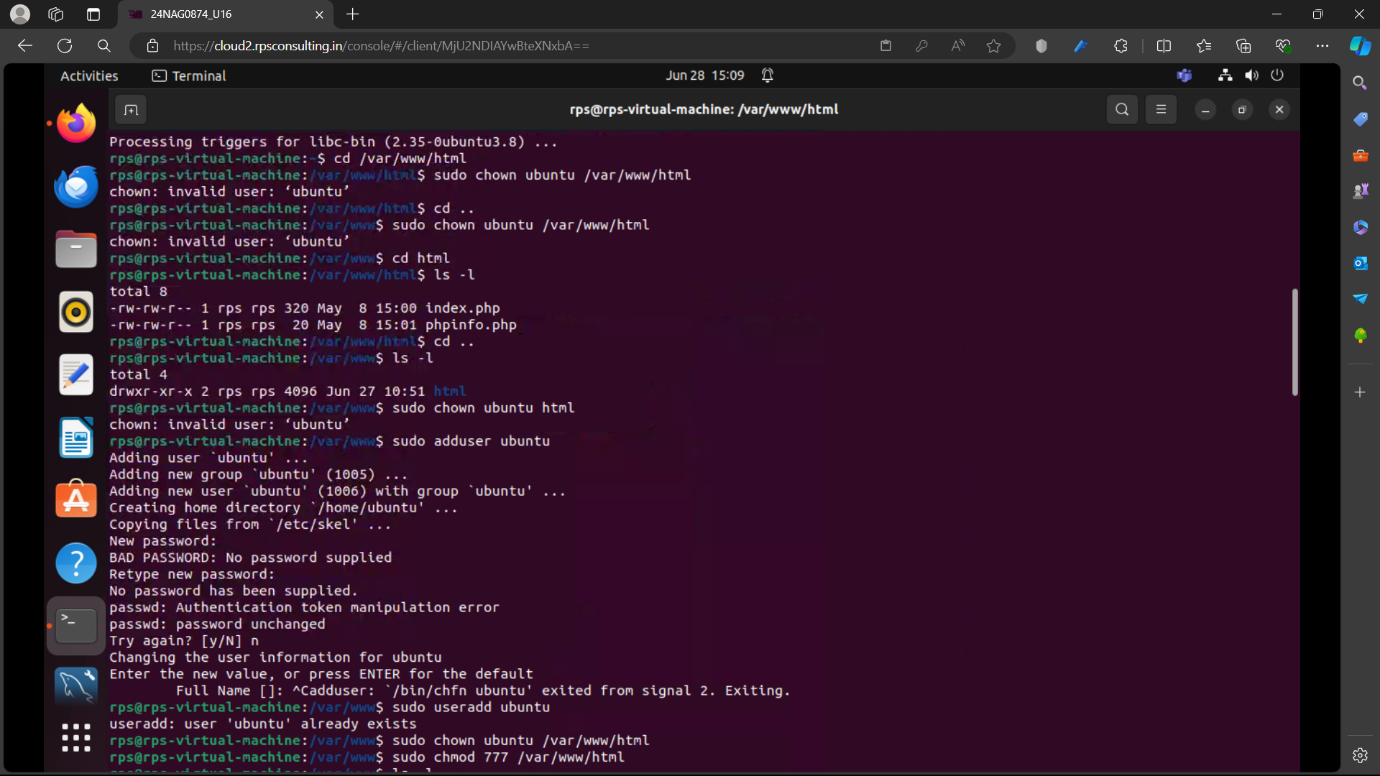


* **Step 2: Configure Apache Document Root**

By default, Apache's web root on Ubuntu is /var/www/html. Ensure it is writable by the user running Apache (typically www-data on Ubuntu), but **avoid using 777 permissions** due to security risks.

sudo chown -R www-data:www-data /var/www/html

sudo chmod -R 755 /var/www/html



* **Step 3: Set Up the Network Scan**

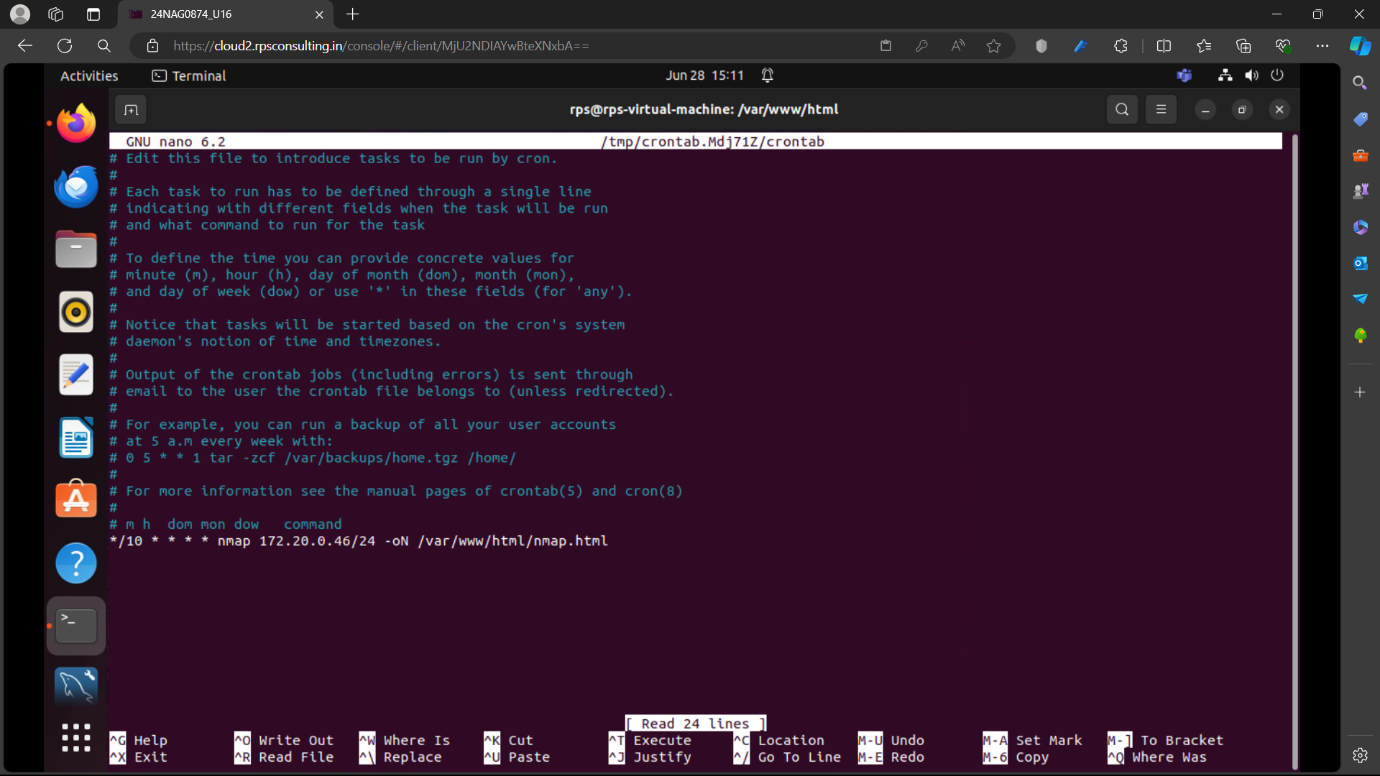
Create a cron job that runs nmap every 10 minutes and saves the output to a file accessible by Apache.

sudo crontab -e

Add the following line to the crontab file:

\*/10 \* \* \* \* nmap 192.168.1.0/24 -oN /var/www/html/nmap.html

This command scans the network 192.168.1.0/24 every 10 minutes and saves the results to /var/www/html/nmap.html.



* **Step 4: Create PHP Script to Display Results**

Create a PHP script (network.php) in /var/www/html that will read the nmap.html file and display its contents along with a timestamp.

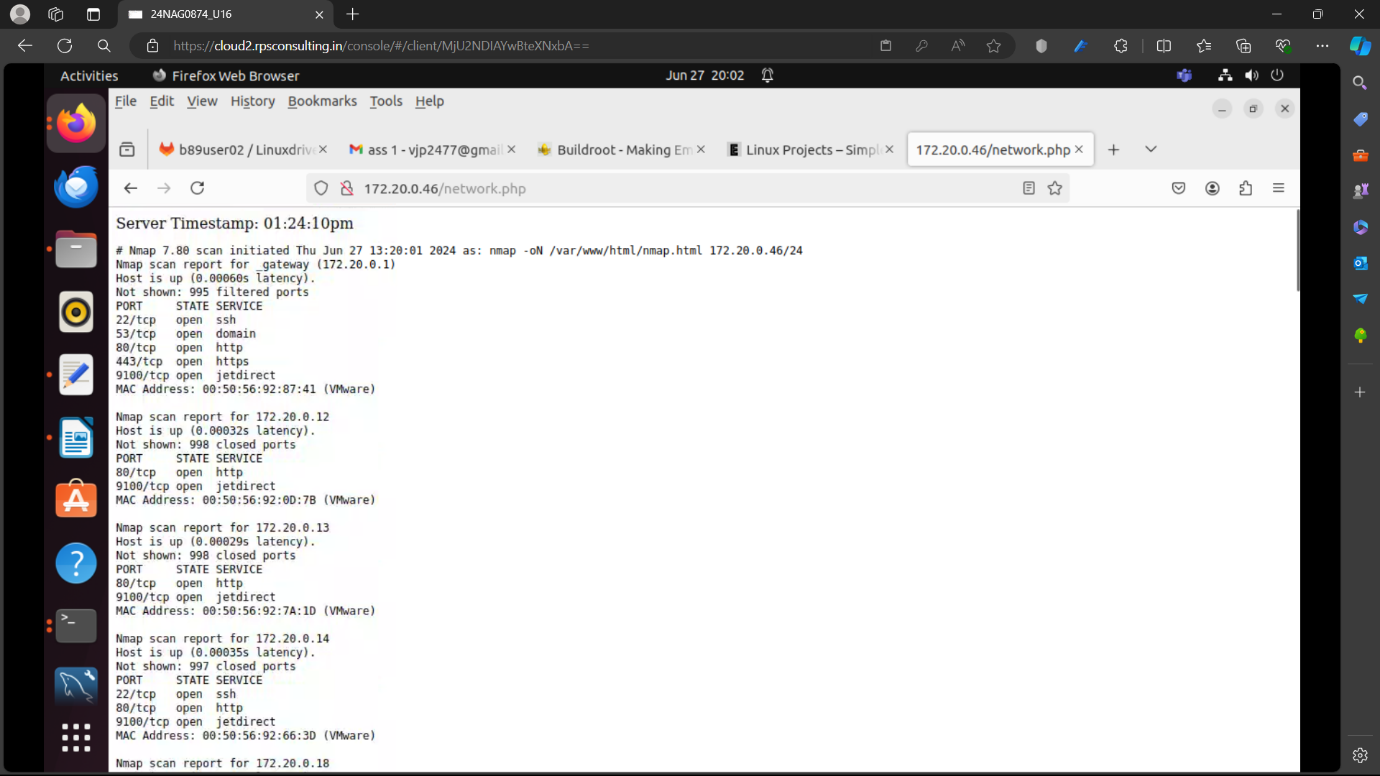
### 

### Step 5: Accessing the Web Interface

Now you can access the network scanner results through a web browser. Assuming your server's IP address is your\_server\_ip, you can access it by navigating to:

http://your\_server\_ip/network.php

This will display the current timestamp and the contents of the nmap.html file, showing the results of the latest network scan.

****

**CONCLUSION**

This project provides a practical solution for network monitoring accessible via a web browser within a local network environment. By integrating cron jobs, nmap, Apache2, and PHP, administrators can automate network scans and conveniently view results through a web interface.

**Key Features:**

* Scheduled Network Scanning: Utilizing cron jobs, the network is scanned every 10 minutes using nmap, targeting the IP range `192.168.1.0/24`. Scan results are saved in a text file (`nmap.html`) within the Apache web server directory (`/var/www/html`).
* Web Interface: The PHP script (`network.php`) displays the current server timestamp and presents formatted scan results from `nmap.html`. This approach allows for easy interpretation of network status directly in a web browser.

**Considerations:**

* Security: Setting directory permissions to `777` (`sudo chmod 777 /var/www/html`) is discouraged in production environments due to potential security vulnerabilities. It's recommended to apply more restrictive permissions based on specific user and group needs.
* Maintenance: Regular monitoring of cron jobs and network scans is essential to ensure ongoing accuracy of network status information presented to administrators.

**Future Directions:**

* Enhanced Visualization\*\*: Implementing graphical representations or interactive elements could improve data visualization and user experience.
* Security Hardening\*\*: Strengthening server security with HTTPS, user authentication, and access controls would safeguard sensitive network information.

In conclusion, this project serves as an effective tool for monitoring and managing local network resources. It combines simplicity with automation, making it suitable for small to medium-sized networks seeking straightforward network scanning and reporting capabilities. Continuous improvement in security practices and user interface enhancements will further elevate its utility in network administration tasks.