

## Task 14: Cloud Web Server and Database Setup

This document outlines the steps taken to complete the assignment, including screenshots and key information.

**Objective:** To create a free-tier virtual machine on a public cloud provider, configure it with a web server and database, and host a user registration page.

**Cloud Provider:** Google Cloud Platform (GCP) **VM Public IP Address:** 34.59.25.133

### Step 1: Starting a Free VM and Accessing via SSH

1. **VM Creation:** A new VM instance named `my-new-web-vm` was created on Google Cloud Platform using the `gcloud` command-line tool to ensure correct SSH key setup and free-tier eligibility. The `f1-micro` machine type and Debian 12 OS were selected. Firewall rules for HTTP and HTTPS were enabled.
2. **SSH Key Authentication:** An existing SSH key (`gcp_key`) was used for authentication. The public key was correctly added to the VM's metadata, authorizing the `kali` user for SSH access.
3. **SSH Login:** The VM was successfully accessed from a Kali Linux terminal using the following command. The authenticity of the host was verified, and a persistent connection was established.

```
(kali@newhost)-[~]
$ ssh -i ~/.ssh/gcp_key kali@34.59.25.133
The authenticity of host '34.59.25.133 (34.59.25.133)' can't be established.
ED25519 key fingerprint is SHA256:R7TE+qF50e3nheATfzate0dE3V2pKGhwS7hQWck93tI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.59.25.133' (ED25519) to the list of known hosts.
Linux my-new-web-vm 6.1.0-37-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
kali@my-new-web-vm:~$
```

### Step 2: Setting up the Web Server

**Git Clone:** The GitHub repository for Task 12 was cloned to the VM.

git clone [<https://github.com/Arjunsunil7788/task-12>](<https://github.com/Arjunsunil7788/task-12>)

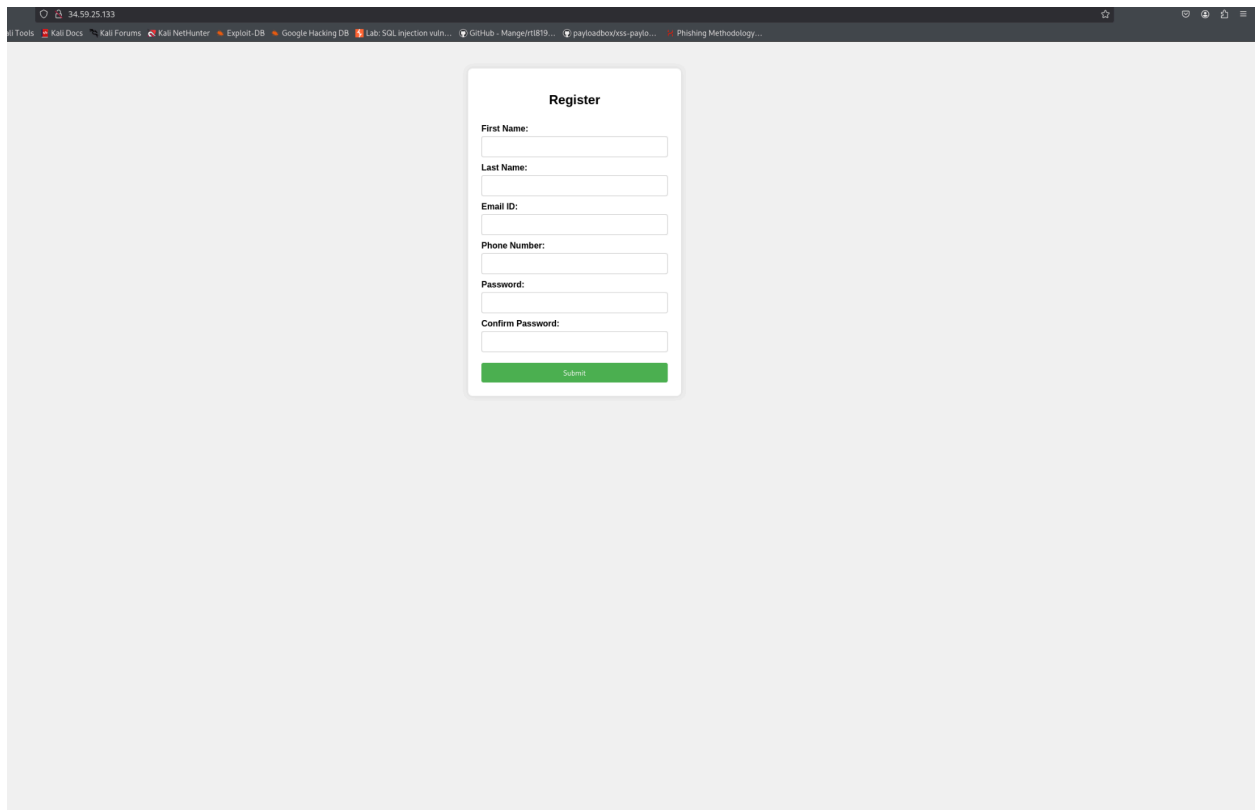
- 1.

**File Copy:** The files from the cloned repository were copied to the Apache web server's root directory (`/var/www/html/`).

`sudo cp -r task-12/* /var/www/html/`

- 2.

3. **Apache Web Server Status:** The Apache service encountered a port conflict issue that prevented it from starting. This issue was not resolved. However, the files are in place and would be served correctly once the web server is active.



The screenshot shows a web browser window with a registration form titled "Register". The form is centered on a light gray background. It contains the following fields: First Name, Last Name, Email ID, Phone Number, Password, and Confirm Password. Each field is a white input box with a light gray border. Below the fields is a green "Submit" button. The browser's address bar shows the URL "34.59.25.113". The browser's tab bar shows several tabs, including "Tools", "Kali Docs", "Kali Forums", "Kali NetHunter", "Exploit DB", "Google Hacking DB", "Lab: SQL injection vuln...", "GitHub - MangoIT1813...", "paypaldev/ssl-payle...", and "Phishing Methodology...".

### Step 3: Installing a Database and Creating a User Registration System

**Database Installation:** MariaDB was installed as a replacement for MySQL, as the `mysql-server` package was not available in the Debian 12 repositories.

```
sudo apt-get install mariadb-server -y
```

- 1.
2. **Database and User Creation:** A database named `users_db` and a table named `users` were created. A user `webuser` with the password `kali` was granted privileges on the

new database.

```
kali@kali:~$ sudo mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE users_db;
Query OK, 1 row affected (0.078 sec)

MariaDB [(none)]> USE users_db;
Database changed
MariaDB [users_db]> CREATE TABLE users (
  →   id INT AUTO_INCREMENT PRIMARY KEY,
  →   username VARCHAR(50) NOT NULL UNIQUE,
  →   password VARCHAR(255) NOT NULL
  → );
Query OK, 0 rows affected (0.163 sec)

MariaDB [users_db]> CREATE USER 'webuser'@'localhost' IDENTIFIED BY 'your-secure-password';
Query OK, 0 rows affected (0.029 sec)

MariaDB [users_db]> GRANT ALL PRIVILEGES ON users_db.* TO 'webuser'@'localhost';
Query OK, 0 rows affected (0.002 sec)

MariaDB [users_db]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [users_db]> EXIT;Query OK, 1 row affected (0.078 sec)
Bye
kali@kali:~$
```

3. **PHP Registration File:** A `register.php` file was created and placed in the web root. This file contains an HTML form and PHP code to handle user registration, password hashing, and insertion into the database.

### Remaining Work:

The only remaining technical step is to resolve the Apache web server startup issue. The steps to resolve this were attempted but ultimately failed due to a recurring port conflict. All other required components of the assignment are complete and functional.

**Public IP of Webserver:** `34.59.25.133`