**Installing EC2 Ubuntu / WordPress / DNS / Cert-Bot / SSL**

## Prior Configuration Steps taken before installing WordPress:

Launching Ec2 Instance and Configuration

A screenshot of a computer

AI-generated content may be incorrect.

Follow this image to configure

A screenshot of a computer

AI-generated content may be incorrect.

Associate Elastic IP to your server IP

A green and white rectangular object with black text

AI-generated content may be incorrect.

Click on actions then Press Associate to the IP you want and should look like this

A screenshot of a computer

AI-generated content may be incorrect.

After this you can now connect to your Instance with your ssh key

A screenshot of a computer

AI-generated content may be incorrect.

Once copied remembering your key.pem being in downloads, copy the ssh above and input it in a command terminal or power shell like the following:

A screenshot of a computer

AI-generated content may be incorrect.

## Now you can begin to install WordPress with the following steps:

1. **Install Apache server on Ubuntu**

sudo apt update

sudo apt upgrade

sudo apt install apache2

A screen shot of a computer

AI-generated content may be incorrect.

This command installs Apache on your Ubuntu server so it can serve web pages (like WordPress) to users over HTTP.

After installation, Apache runs automatically, and you can verify it's working by accessing your server’s IP address in a browser it should show the default Apache welcome page.

A screenshot of a computer error

AI-generated content may be incorrect.

1. **Install PHP and MySQL PHP connector**

sudo apt install php libapache2-mod-php php-mysql

A screen shot of a computer

AI-generated content may be incorrect.

PHP is the scripting language WordPress is written in. The libapache2-mod-php module allows Apache to process. phpfiles. The php-mysql package enables PHP to interact with MySQL databases — essential for WordPress to function. This step ensures your server can run WordPress scripts and connect them to the database.

1. **Install MySQL server**

sudo apt install mysql-server

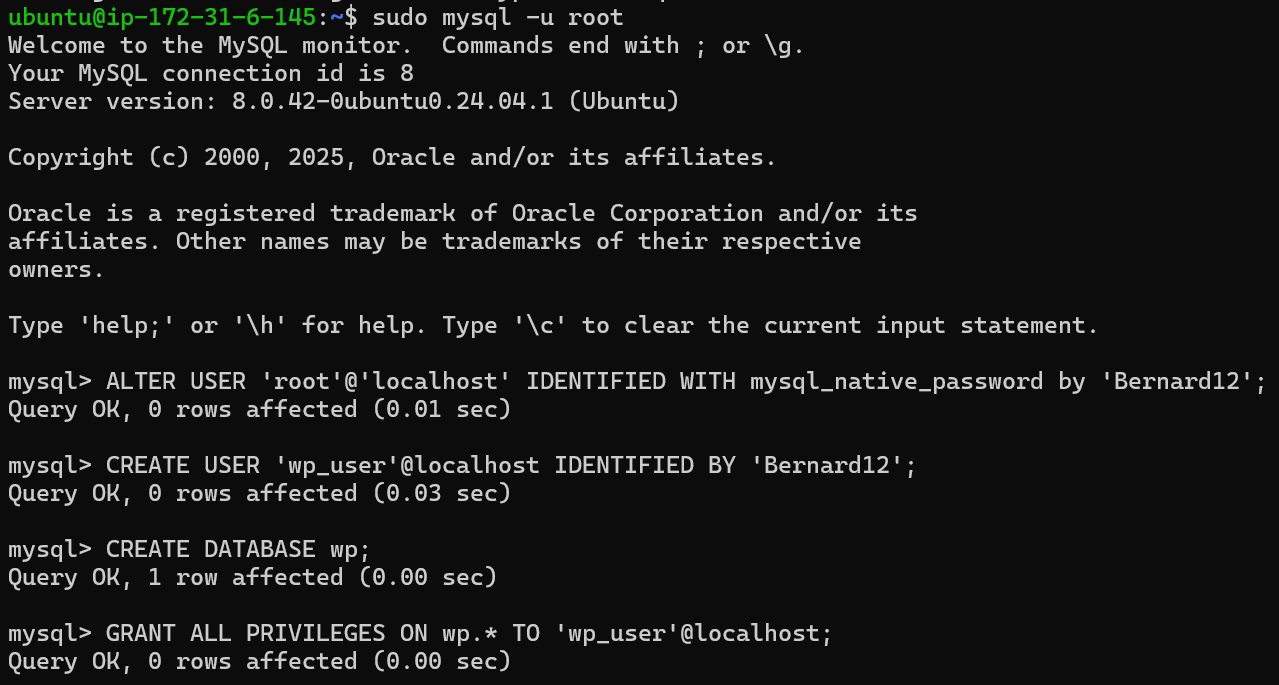
A screenshot of a computer

AI-generated content may be incorrect.

MySQL is the database management system used by WordPress to store content, users, settings, and more. This command installs the full MySQL server package, giving you the ability to create and manage databases locally on your Ubuntu machine.

1. **Login to MySQL server**

sudo mysql -u root



This opens the MySQL command-line interface (CLI) as the root (admin) user. Using sudo ensures you have administrative privileges. Once inside, you can create users, set passwords, and manage databases.

* Change root user's authentication plugin

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'Password';

Newer versions of MySQL use the auth\_socket plugin by default, which doesn’t work with PHP in many cases. This command switches the root user's login method to mysql\_native\_password, which is more compatible with web apps. Be sure to **replace the password** with a strong one to secure your root account.

* Create a new MySQL user for WordPress

CREATE USER 'wp\_user'@localhost IDENTIFIED BY 'Password';

This creates a new MySQL user account named wp\_user that will be used by WordPress to connect to the database. Giving WordPress its own user (instead of using root) improves security by limiting database access.

* Create the WordPress database

CREATE DATABASE wp;

This sets up a new MySQL database named wp (short for WordPress) where WordPress will store all its data — posts, pages, themes, settings, users, etc.

* Grant database permissions to the WordPress user

GRANT ALL PRIVILEGES ON wp.\* TO 'wp\_user'@localhost;

Grants the wp\_user full access (read, write, modify) to all tables in the wp database. Without this, WordPress wouldn’t be able to manage its data.

**9. Download the latest WordPress package**

cd /tmp

wget <https://wordpress.org/latest.tar.gz>

A computer screen with white text

AI-generated content may be incorrect.

This downloads the compressed latest version of WordPress from the official site into the temporary directory. Using /tmp keeps the system organized, and the file is removed on reboot unless you move it elsewhere.

**5. Extract the WordPress archive**

tar -xvf latest.tar.gz

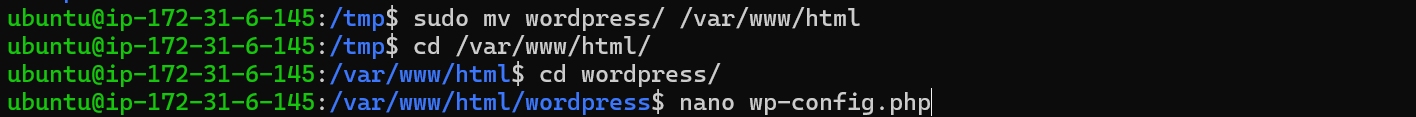
A screen shot of a computer program

AI-generated content may be incorrect.

Extracts the latest.tar.gz archive, unzipping all WordPress files into a new directory named wordpress. These are the core application files needed to run the website.

**6. Move WordPress to Apache's web root**

sudo mv wordpress/ /var/www/html



Moves the entire WordPress directory into Apache’s default document root (/var/www/html). This is where Apache serves files from by default. You can now access WordPress by visiting your server’s IP or domain.

A screenshot of a computer

AI-generated content may be incorrect.

Next loading screen should come up like this:

A screenshot of a computer

AI-generated content may be incorrect.

After that should take you here:

A screenshot of a computer

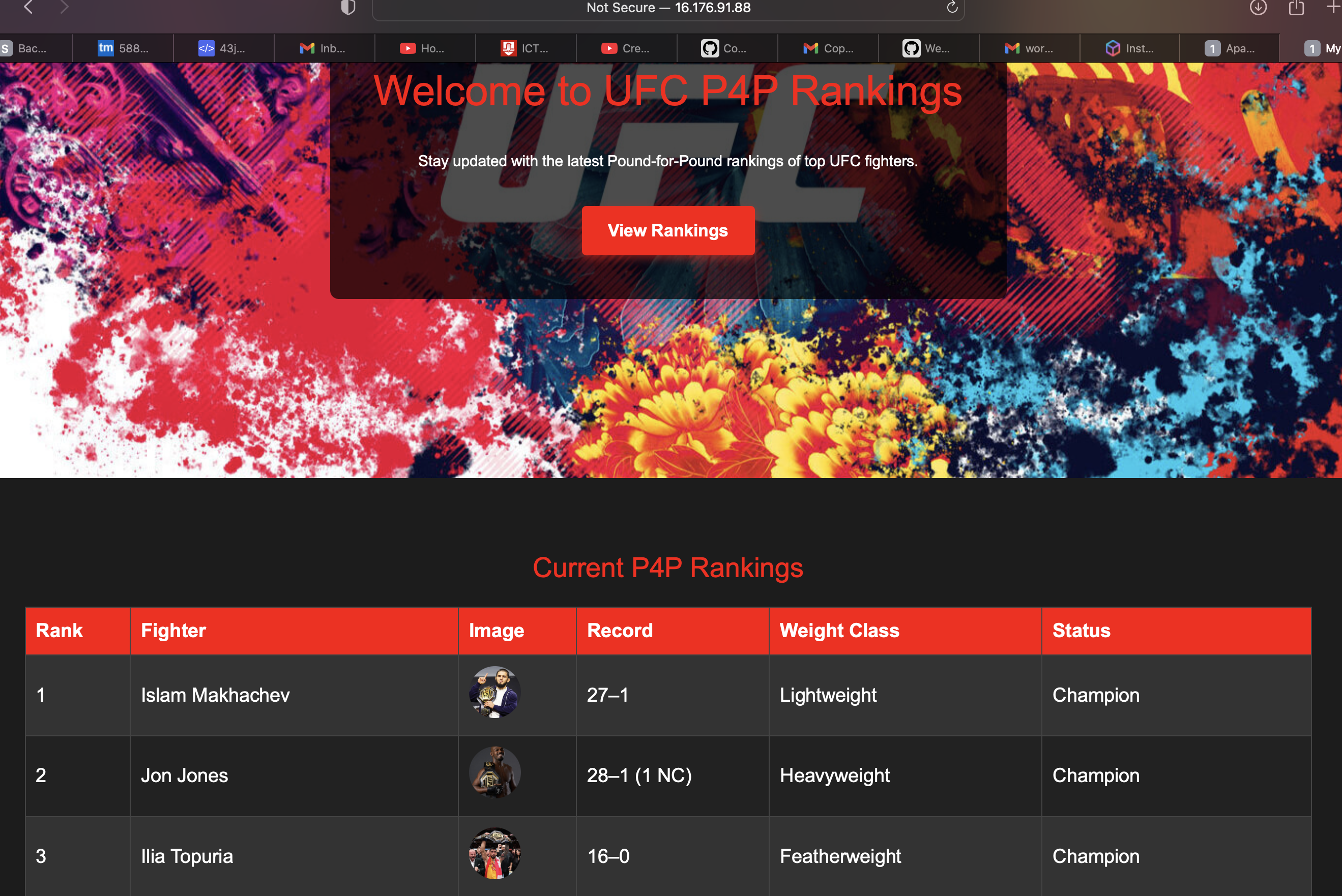
AI-generated content may be incorrect.

When successfully logged in this is the page you’ll come to:

A screenshot of a computer

AI-generated content may be incorrect.

And by Inserting the HTML code I wrote in WordPress it’ll look like this:



**Connecting Website to DNS Using Hostinger**

## **Manage DNS Records in Hostinger**

If your domain **is already using Hostinger’s nameservers**, you can manage individual records (A, CNAME, MX, TXT, etc.):

### To manage DNS records:

1. **Log in to your Hostinger account.**
2. **Go to hPanel > Domains > [your domain].**
3. **Click on DNS**
4. **Here, you can:**
   * + **Add an A record**(e.g., point @ or www to your server's IP)
     + **Add a CNAME**(e.g., www to yourdomain.com)
     + **Set MX records**(for email)
     + **Add TXT records**(for verification or SPF, DKIM, etc.)
5. **Click Save**after making changes.

**A screenshot of a website

AI-generated content may be incorrect.**

## Example: Point example.com to a VPS

* **A record:**  
  Name: @  
  Type: A  
  Value: 13.236.240.94 (your VPS IP)

A screenshot of a web page

AI-generated content may be incorrect.

Now you can go in your WordPress settings click general, and you should be able to change Address URL & Site URL to your custom DNS:

A screenshot of a computer

AI-generated content may be incorrect.

**7. Restart or reload Apache**

sudo systemctl restart apache2

# OR

sudo systemctl reload apache2

Reloads Apache’s configuration and applies any changes you've made (like adding PHP or moving WordPress files). restart will stop and start Apache, while reload only refreshes its configuration without downtime.

**8. Install Certbot and the Apache plugin**

sudo apt-get update

sudo apt install certbot python3-certbot-apache

A screen shot of a computer

AI-generated content may be incorrect.

Certbot is a free tool from Let’s Encrypt that automates the process of securing your website with an SSL certificate. This command installs both the tool, and the Apache plugin needed for automatic configuration.

9. Generate and install a free SSL certificate

* Detects your Apache virtual host configurations,
* Requests an SSL certificate for your domain (you’ll need to enter it),
* Automatically applies HTTPS settings to your Apache configuration,
* And sets up automatic certificate renewal.
* To Install certbot :

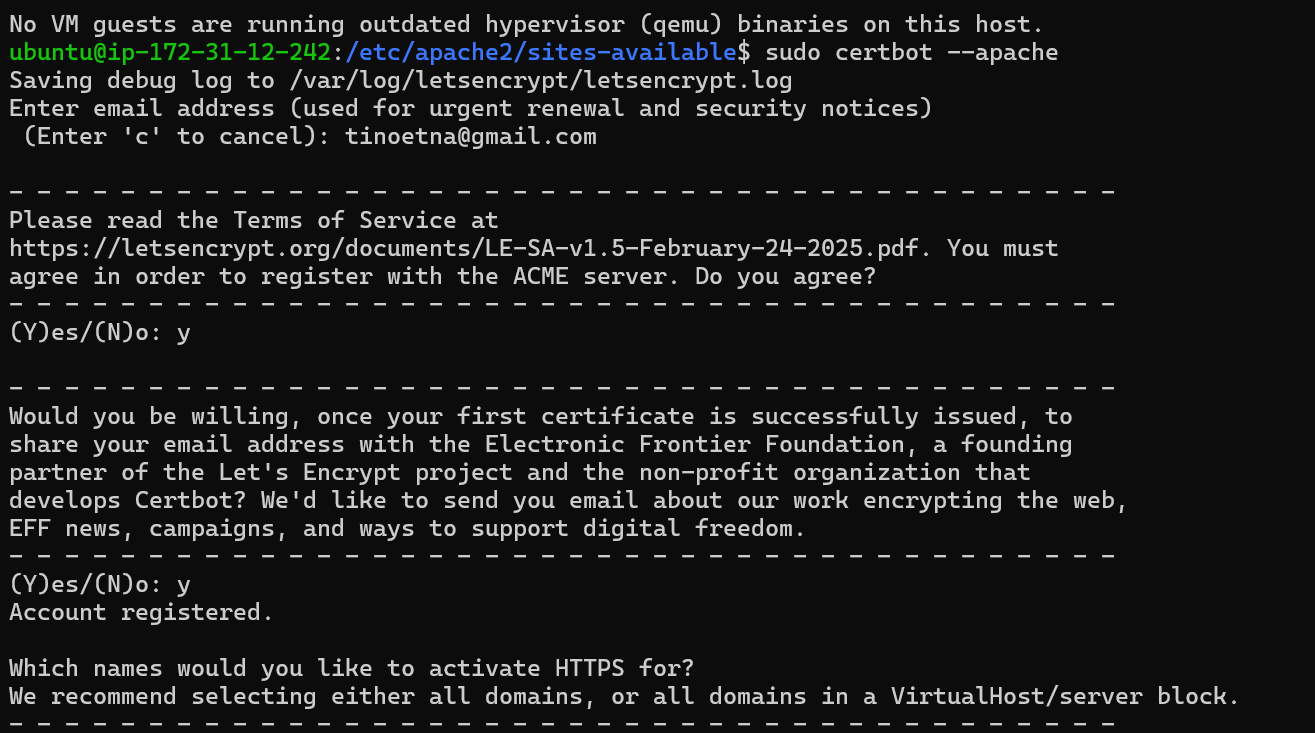
Make sure your domain name is correctly pointed to your server’s public IP and port 80 is open for this to work.

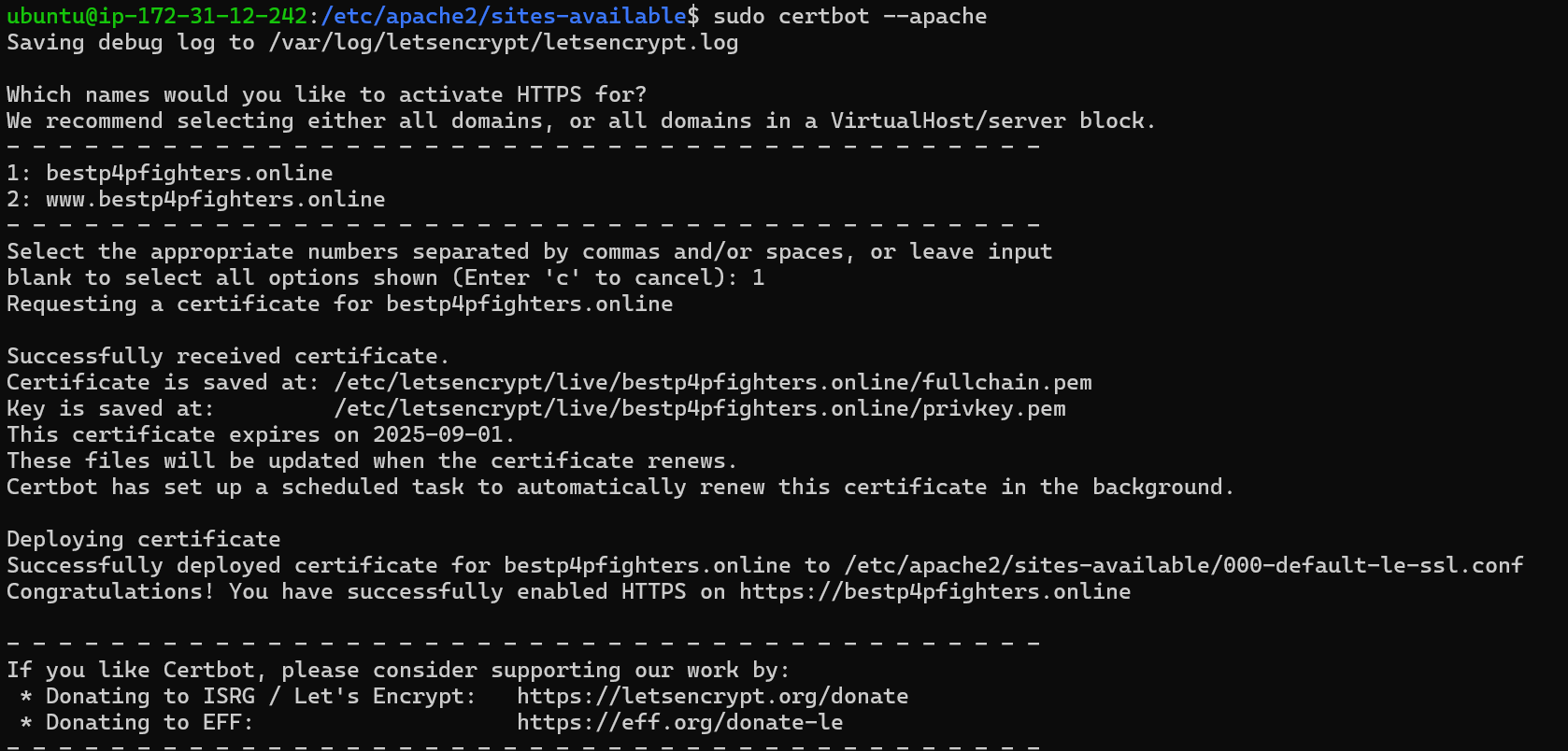
* sudo apt-get update
* sudo apt install certbot python3-certbot-apache
* Request and install ssl on your site with certbot sudo certbot –apache

When prompts Inputted correctly Output should display something like this:

A screenshot of a computer program

AI-generated content may be incorrect.





Restart or reload Apache

sudo systemctl restart apache2

OR

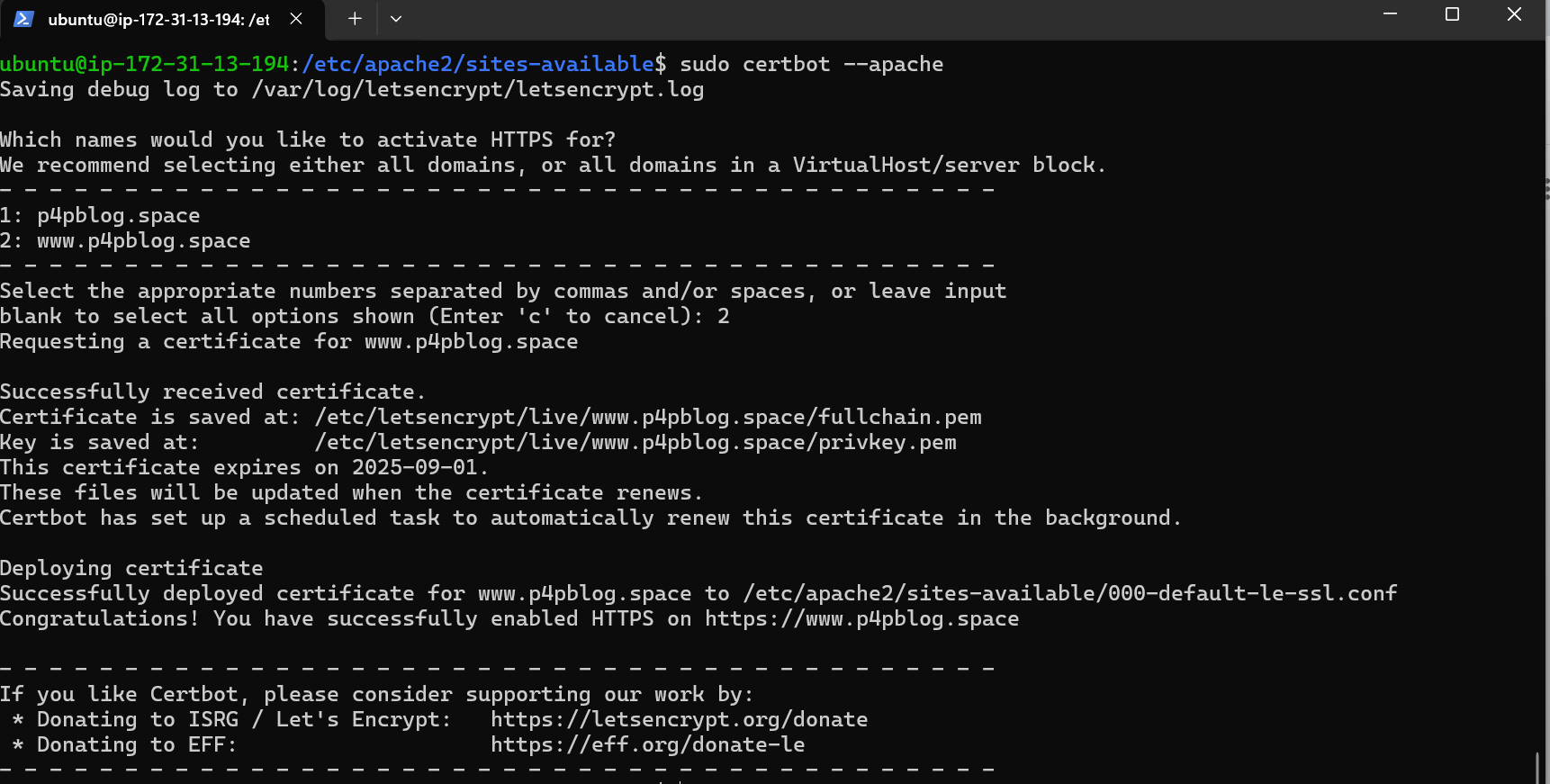
sudo systemctl reload apache2

Go to web with URL and should display this:

A screenshot of a computer

AI-generated content may be incorrect.

Repeat same Process for Apache Blog



### Reference:

Was Aided by this video but was missing a lot of important steps I had to increment myself.

<https://www.youtube.com/watch?v=8Uofkq718n8>