

Installing WordPress/Cert-Bot/SSL

Prior Configuration Steps taken before installing WordPress:

Launching Ec2 Instance and Configuration

Launch an instance [Info](#)
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)
Name
ICT171-UFC-Web [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents **Quick Start**

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

SUSE Linux

Debian

Amazon Machine Image (AMI)
Ubuntu Server 24.04 LTS (HVM), SSD Volume Type
ami-0f5d1713c9a4fe30 (i64-bit (x86_64)) / ami-099eab5b168040255 (i64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Summary
Number of instances [Info](#)
1
Software Image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0f5d1713c9a4fe30
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

Follow this image to configure

Key pair (login) [Info](#)
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
FinalP [Create new key pair](#)

Network settings [Info](#) [Edit](#)
Network [Info](#)
vpc-0eac3c3ab3fc361fe
Subnet [Info](#)
No preference (Default subnet in any availability zone)
Auto-assign public IP [Info](#)
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
☒ Create security group ☐ Select existing security group
We'll create a new security group called 'launch-wizard-10' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance Anywhere
0.0.0.0/0

☒ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Associate Elastic IP to your server IP

Elastic IP address allocated successfully.
Elastic IP address 16.176.91.88

[Associate this Elastic IP address](#)

Elastic IP addresses (1) [Info](#)

Find elastic IP addresses by attribute or tag

Public IPv4 address: 16.176.91.88 [X](#) [Clear filters](#)

<input type="checkbox"/>	Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID	Private IP address
<input type="checkbox"/>	-	16.176.91.88	Public IP	eipalloc-0400768f651345b14	-	-	-

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

Associate Elastic IP address [Info](#)

Choose the instance or network interface to associate to this Elastic IP address (16.176.91.88)

Elastic IP address: 16.176.91.88

Resource type
Choose the type of resource with which to associate the Elastic IP address.

☒ Instance
☐ Network interface

Instance
Choose an instance

i-0dd56c1c592ed0fb5 (ICT171-UFC-Web) - running

Reassociation
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.

☐ Allow this Elastic IP address to be reassociated

[Cancel](#) [Associate](#)

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

Connect to instance [Info](#)

Connect to your instance i-0dd56c1c592ed0fb5 (ICT171-UFC-Web) using any of these options

EC2 Instance Connect | **Session Manager** | **SSH client** | **EC2 serial console**

Instance ID
i-0dd56c1c592ed0fb5 (ICT171-UFC-Web)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is FinalP.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "FinalP.pem"`
4. Connect to your instance using its Public DNS:
`ec2-13-236-187-29.ap-southeast-2.compute.amazonaws.com`

Example:
`ssh -i "FinalP.pem" ubuntu@ec2-13-236-187-29.ap-southeast-2.compute.amazonaws.com`

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:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\bChik> cd downloads
PS C:\Users\bChik\downloads> ssh -i "FinalP.pem" ubuntu@ec2-16-176-91-88.ap-southeast-2.compute.amazonaws.com
```

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

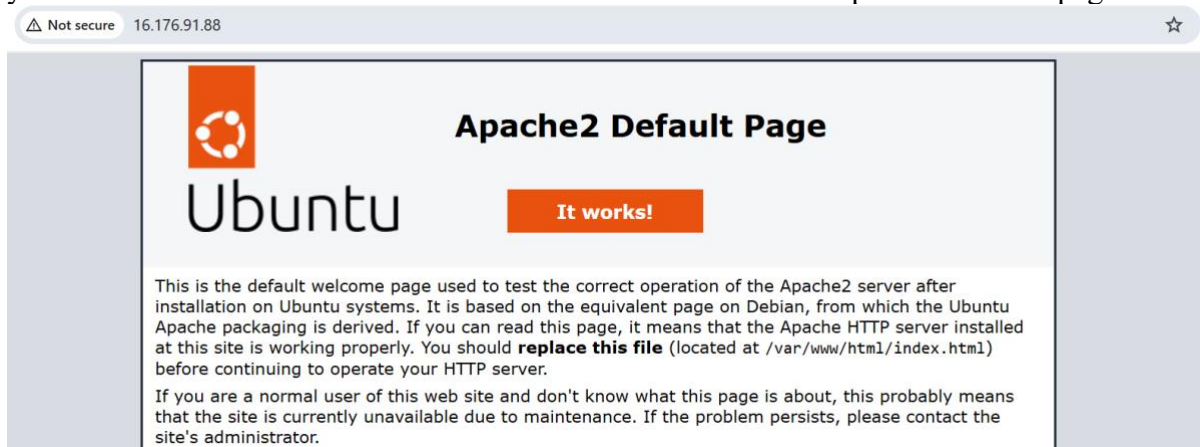
1. Install Apache server on Ubuntu

```
sudo apt update
sudo apt upgrade
sudo apt install apache2
```

```
ubuntu@ip-172-31-6-145:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

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.



2. Install PHP and MySQL PHP connector

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

```
ubuntu@ip-172-31-6-145:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.3 php-common php8.3 php8.3-cli php8.3-common php8.3-mysql php8.3-opcache php8.3-readline
  php-pear
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.3 php php-common php-mysql php8.3 php8.3-cli php8.3-common php8.3-mysql
  php8.3-opcache php8.3-readline
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 5050 kB of archives.
After this operation, 22.9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

PHP is the scripting language WordPress is written in. The libapache2-mod-php module allows Apache to process .php files. 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.

3. Install MySQL server

```
sudo apt install mysql-server
```

```
ubuntu@ip-172-31-6-145:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin
  libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64 libtimedate-perl
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common
  mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl
  libwww-perl mailx tinyc
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl libevent-pthreads-2.1-7t64 libfcgi-bin
  libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl
  libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite32t64 libtimedate-perl
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common
  mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.6 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

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.

4. Login to MySQL server

```
sudo mysql -u root
```

```
ubuntu@ip-172-31-6-145:~$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.42-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'Bernard12';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'wp_user'@localhost IDENTIFIED BY 'Bernard12';
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE DATABASE wp;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON wp.* TO 'wp_user'@localhost;
Query OK, 0 rows affected (0.00 sec)
```

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
```

```
ubuntu@ip-172-31-6-145:~$ cd /tmp
ubuntu@ip-172-31-6-145:/tmp$ wget https://wordpress.org/latest.tar.gz
--2025-05-27 02:30:03-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26926501 (26M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz          100%[=====] 25.68M  143KB/s  in 2m 55s
2025-05-27 02:32:59 (150 KB/s) - 'latest.tar.gz' saved [26926501/26926501]
```

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
```

```
ubuntu@ip-172-31-6-145:/tmp$ ls
latest.tar.gz
snap-private-tmp
systemd-private-b46e79a2ef2747f29a20dd02132b484a-ModemManager.service-YvGV0U
systemd-private-b46e79a2ef2747f29a20dd02132b484a-apache2.service-yGfTGc
systemd-private-b46e79a2ef2747f29a20dd02132b484a-chrony.service-QGRj83
systemd-private-b46e79a2ef2747f29a20dd02132b484a-polkit.service-ZFpASz
systemd-private-b46e79a2ef2747f29a20dd02132b484a-systemd-logind.service-uYFwh8
systemd-private-b46e79a2ef2747f29a20dd02132b484a-systemd-resolved.service-023VrY
tmp.H2dcjvL6aS
tmp.HKksuH4qe1
wordpress
```

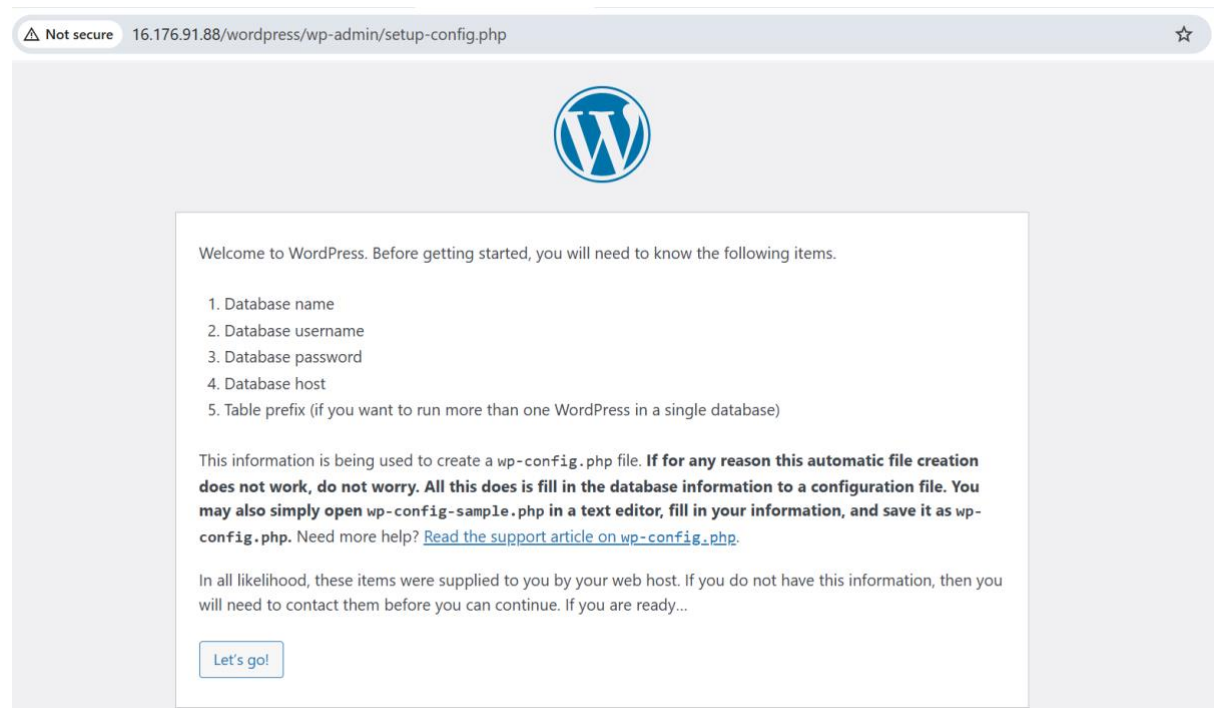
Extracts the latest.tar.gz archive, unzipping all WordPress files into a new directory named wordpress. These are the core application files needed to run the website.

6. Move WordPress to Apache's web root

```
sudo mv wordpress/ /var/www/html
```

```
ubuntu@ip-172-31-6-145:/tmp$ sudo mv wordpress/ /var/www/html
ubuntu@ip-172-31-6-145:/tmp$ cd /var/www/html/
ubuntu@ip-172-31-6-145:/var/www/html$ cd wordpress/
ubuntu@ip-172-31-6-145:/var/www/html/wordpress$ nano wp-config.php
```

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.



Next loading screen should come up like this:



Below you should enter your database connection details. If you are not sure about these, contact your host.

Database Name
The name of the database you want to use with WordPress.

Username
Your database username.


Password [Show](#)
Your database password.

Database Host
You should be able to get this info from your web host, if `localhost` does not work.

Table Prefix
If you want to run multiple WordPress installations in a single database, change this.

After that should take you here:

16.176.91.88/wordpress/wp-admin/install.php?step=2



Success!

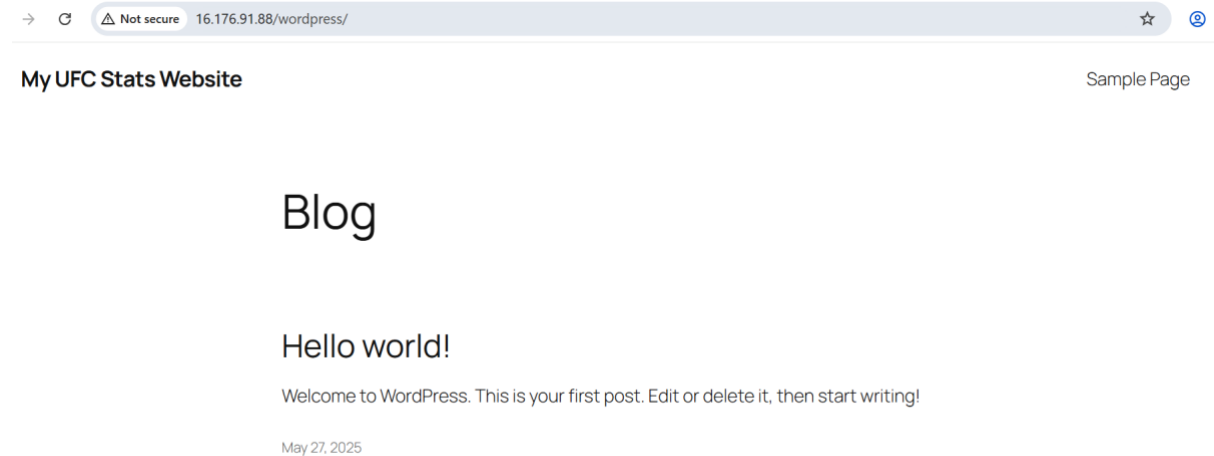
WordPress has been installed. Thank you, and enjoy!

Username Wesley_t_c

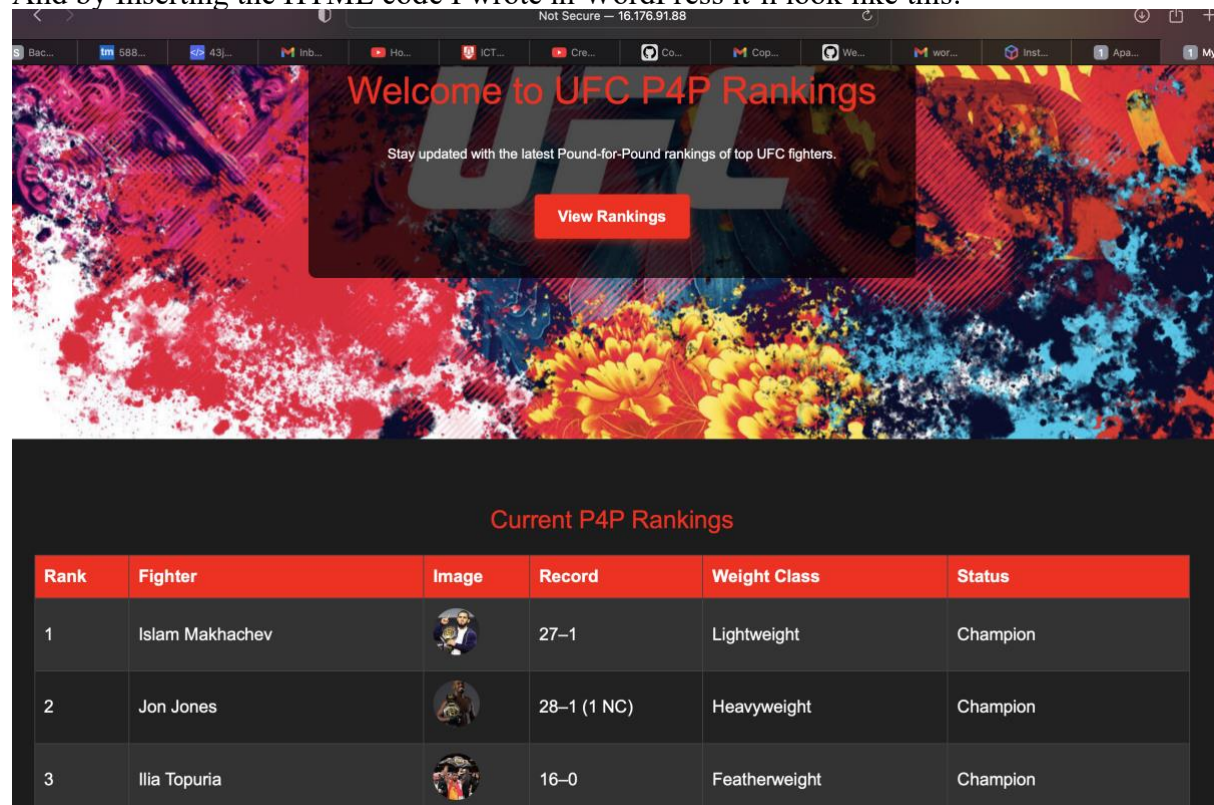
Password *Your chosen password.*

[Log In](#)

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



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



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
```

```
ubuntu@ip-172-31-6-145:/etc/apache2/sites-available$ sudo apt-get update
Hit:1 http://ap-southeast-2-ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-southeast-2-ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-southeast-2-ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [21.5 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.2 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Fetched 200 kB in 3s (77.6 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-6-145:/etc/apache2/sites-available$ sudo apt install certbot python3-certbot-apache
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  Augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot python3-configargparse python3-icu
  python3-josepy python3-parsedatetime python3-rfc3339
Suggested packages:
  Augeas-doc python-certbot-doc python3-certbot-nginx Augeas-tools python-acme-doc python-certbot-apache-doc
The following NEW packages will be installed:
  Augeas-lenses certbot libaugeas0 python3-acme python3-augeas python3-certbot python3-certbot-apache
  python3-configargparse python3-icu python3-josepy python3-parsedatetime python3-rfc3339
0 upgraded, 12 newly installed, 0 to remove and 0 not upgraded.
Need to get 1657 kB of archives.
After this operation, 8599 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

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

```
sudo certbot --apache
```

- ❖ 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.

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

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>

