

# Hosting a WordPress Website on Amazon EC2

This document explains how to use Amazon EC2 to host a WordPress and connect it to a custom domain using Apache webserver and MySQL database.

**Following are the Steps to host a wordpress website on Amazon EC2 with custom domain:**

Step 1: Setup Amazon EC2 instance

Step 2: Install apache webserver

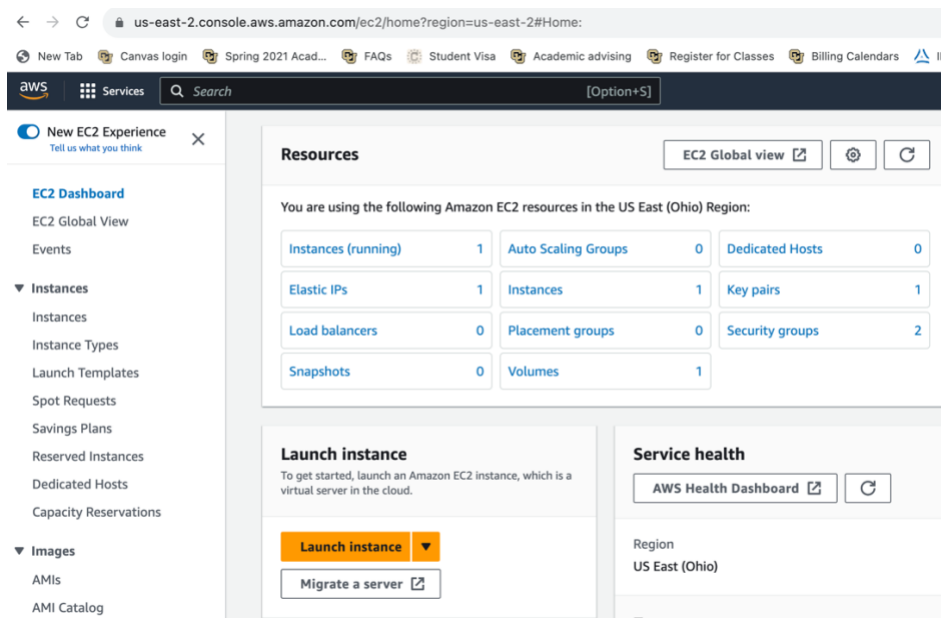
Step 3: Install MySQL database

Step 4: Download and install wordpress

Step 5: Link a custom domain to a website

## Step1: Setup Amazon EC2 instance

Open Amazon EC2 console



Click on “Launch instance” and enter an instance name

Name : My wordpress server

Application and OS Images (Amazon Machine Image) : Ubuntu

**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  
 [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 | macOS | **Ubuntu** | Windows | Red Hat | SUSE Linux

Instance type : t2.micro

Key pair: wordpress\_server , others as default click on “create key pair”

**Create key pair** ×

**Key pair name**  
 Key pairs allow you to connect to your instance securely.  
  
 The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

☒ **RSA**  
 RSA encrypted private and public key pair

☐ **ED25519**  
 ED25519 encrypted private and public key pair

**Private key file format**

☒ **.pem**  
 For use with OpenSSH

☐ **.ppk**  
 For use with PuTTY

**⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)**

Cancel Create key pair

This downloads the wordpress\_server.pem file (Secure file)

Under network settings, enable all the three traffic

Allow SSh traffic from

Allow HTTPs traffic from the internet

Allow HTTP traffic from the internet

The screenshot shows the 'Network settings' section of the AWS console. It includes fields for 'Network' (vpc-0ec1fcb71dfc66140) and 'Subnet' (No preference). The 'Auto-assign public IP' option is set to 'Enable'. Under 'Firewall (security groups)', the 'Create security group' radio button is selected. Below this, it states that a new security group named 'launch-wizard-2' will be created with three default rules: 'Allow SSH traffic from Anywhere', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. A yellow warning box at the bottom states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

Others as default and finally click on “Launch instance”

The screenshot shows the 'Configure storage' section of the AWS console. It displays '1x 8 GiB gp2 Root volume (Not encrypted)'. A blue information box states: 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. There is an 'Add new volume' button. Below this, it notes: 'The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance'. At the bottom, there are sections for 'Advanced details' and 'Summary', and two buttons: 'Cancel' and 'Launch instance'.

Once the instance is created, now assign the Elastic Ip address in order to have constant public IPV4 address.

Goto EC2 dashboard, click on “Elastic IPs”, click on “Allocate Elastic IP address”, and click on “Allocate”. Now created Elastic IP address.

aws Services Search [Option+S]

**Amazon's pool of IPv4 addresses**

- ☐ Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) [Learn more](#)
- ☐ Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) [Learn more](#)

**Global static IP addresses**

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

[Create accelerator](#)

**Tags - optional**

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tag

Cancel [Allocate](#)

Select the Elastic IP address created and Goto Actions and select the Associate Elastic IP address. Here choose an EC2 instance and click on “Associate”.

aws Services Search [Option+S]

**Resource type**

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

☒ Instance

☐ Network interface

**Warning**

If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, but the address will still be allocated to your account. [Learn more](#)

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

**Instance**

**Private IP address**

The private IP address with which to associate the Elastic IP address.

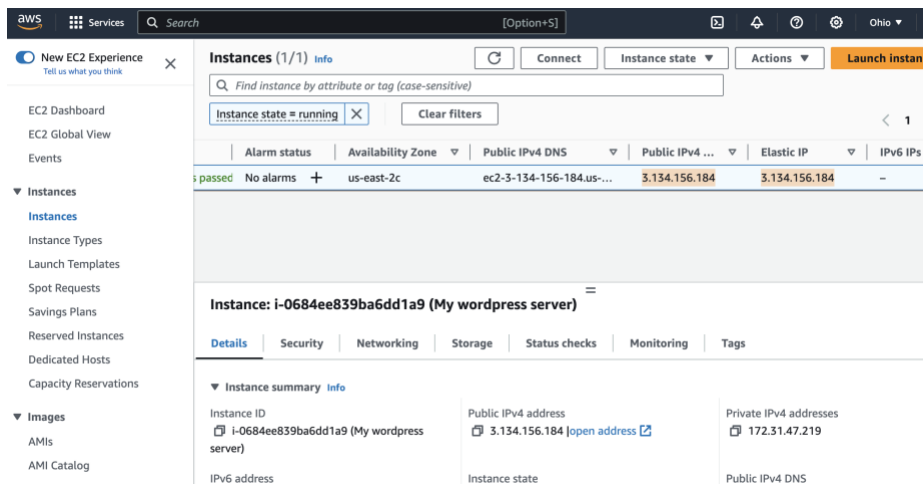
**Reassociation**

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](#)

Now you can see public IPV4 address is changed and is same as Elastic IP.



Open the SSH client (terminal)

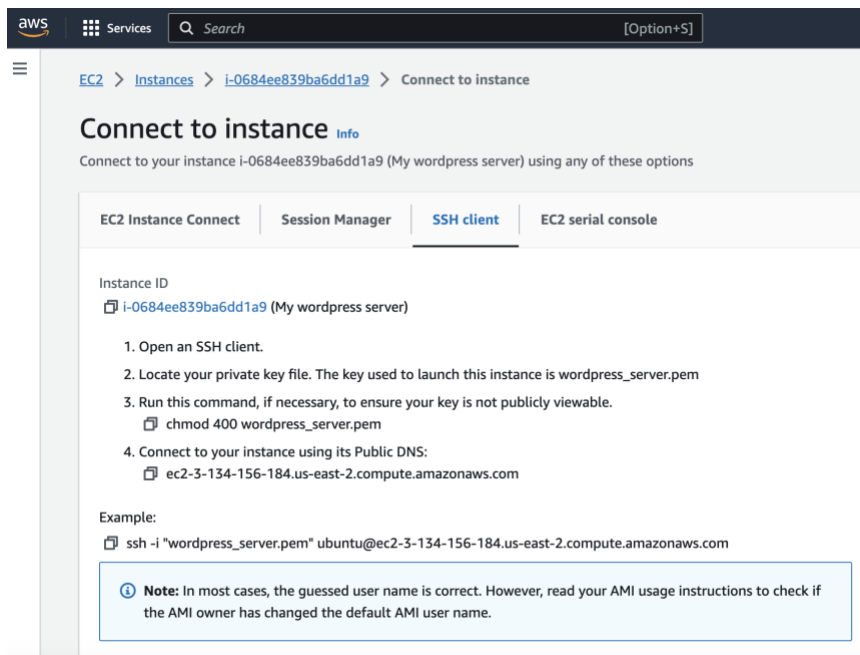
Locate your private key file. The key used to launch this instance is `wordpress_server.pem`

Run this command, if necessary, to ensure your key is not publicly viewable.

`chmod 400 wordpress_server.pem`

Right click on the instance created, click on “connect”.

Under SSH client section, there is command to connect through SSH.



## Connect to your instance in SSH client

```
ssh -i "wordpress_server.pem" ubuntu@ec2-3-134-156-184.us-east-2.compute.amazonaws.com
```

```
[mallika:Downloads sharathkrishna$ ssh -i "wordpress_server.pem" ubuntu@ec2-3-134-156-184.us-east-2.compute.amazonaws.com
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)
```

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage
```

System information as of Mon Oct 2 18:27:48 UTC 2023

```
System load:  0.0          Processes:      107
Usage of /:   38.9% of 7.57GB Users logged in:  0
Memory usage: 71%         IPv4 address for eth0: 172.31.47.219
Swap usage:   0%
```

\* Ubuntu Pro delivers the most comprehensive open source security and compliance features.

<https://ubuntu.com/aws/pro>

Expanded Security Maintenance for Applications is not enabled.

63 updates can be applied immediately.

To see these additional updates run: `apt list --upgradable`

Enable ESM Apps to receive additional future security updates.

See <https://ubuntu.com/esm> or run: `sudo pro status`

\*\*\* System restart required \*\*\*

Last login: Thu Sep 28 12:40:02 2023 from 76.25.165.0

ubuntu@ip-172-31-47-219:~\$

## Now you are connected to your instance.

## Step2: Install Apache Webserver

```
sudo apt install apache2
```

```
ubuntu@ip-172-31-47-219:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 65 not upgraded.
Need to get 97.8 kB of archives.
After this operation, 546 kB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.6 [97.8 kB]
Fetched 97.8 kB in 0s (5835 kB/s)
Selecting previously unselected package apache2.
(Reading database ... 95546 files and directories currently installed.)
Preparing to unpack .../apache2_2.4.52-1ubuntu4.6_amd64.deb ...
Unpacking apache2 (2.4.52-1ubuntu4.6) ...
Setting up apache2 (2.4.52-1ubuntu4.6) ...
apache-htcacheclean.service is a disabled or a static unit not running, not starting it.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for ufw (0.36.1-4build1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Restarting services...
/etc/needrestart/restart.d/systemd-manager
systemctl restart chrony.service multipathd.service mysql.service packagekit.service polkit.service rsyslog.service ssh.service systemd-journald.service systemd-networkd.service systemd-resolved.service
systemd-udev.service
Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service
systemctl restart user@1000.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-47-219:~$
```

Cross Verify if the webserver is working, In the browser open the public IPV4 address and check the apache2 default page.



**Apache2 Default Page**

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
-- apache2.conf
-- ports.conf
-- mods-enabled
-- conf-enabled
-- sites-enabled
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2dissconf`. See their respective man pages for detailed information.
- The binary is called `apache2` and is managed using `systemd`, so to start/stop the service use

Now install php runtime on our instance and MySQL connector for php.

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

```
ubuntu@ip-172-31-47-219:~$ 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-php8.1 php-common php8.1-cgi php8.1-common
  php8.1-mysql php8.1-opcache php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.1 php-common php-mysql php8.1
  php8.1-cgi php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
0 upgraded, 11 newly installed, 0 to remove and 66 not upgraded.
Need to get 5265 kB of archives.
After this operation, 21.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-1ubuntu2.14 [1127 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2.14 [365 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2.14 [13.6 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cgi amd64 8.1.2-1ubuntu2.14 [1834 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapache2-mod-php8.1 amd64 8.1.2-1ubuntu2.14 [1766 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapache2-mod-php all 2:8.1+92ubuntu1 [2899 B]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1 all 8.1.2-1ubuntu2.14 [9158 B]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php all 2:8.1+92ubuntu1 [2758 B]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-mysql amd64 8.1.2-1ubuntu2.14 [130 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-mysql all 2:8.1+92ubuntu1 [1834 B]
Fetched 5265 kB in 4s (1451 MB/s)
Selecting previously unselected package php-common.
(Reading database ... 94076 files and directories currently installed.)
Preparing to unpack .../00-php-common_2:92ubuntu1_all.deb ...
Unpacking php-common (2:92ubuntu1) ...
Selecting previously unselected package php8.1-common.
Preparing to unpack .../01-php8.1-common_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking php8.1-common (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package php8.1-opcache.
Preparing to unpack .../02-php8.1-opcache_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking php8.1-opcache (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package php8.1-readline.
Preparing to unpack .../03-php8.1-readline_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking php8.1-readline (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package php8.1-cgi.
Preparing to unpack .../04-php8.1-cgi_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking php8.1-cgi (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package libapache2-mod-php8.1.
Preparing to unpack .../05-libapache2-mod-php8.1_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking libapache2-mod-php8.1 (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package libapache2-mod-php.
Preparing to unpack .../06-libapache2-mod-php_2:8.1+92ubuntu1_all.deb ...
Unpacking libapache2-mod-php (2:8.1+92ubuntu1) ...
Selecting previously unselected package php8.1.
Preparing to unpack .../07-php8.1_8.1.2-1ubuntu2.14_all.deb ...
Unpacking php8.1 (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package php.
Preparing to unpack .../08-php_2:8.1+92ubuntu1_all.deb ...
Unpacking php (2:8.1+92ubuntu1) ...
Selecting previously unselected package php8.1-mysql.
Preparing to unpack .../09-php8.1-mysql_8.1.2-1ubuntu2.14_amd64.deb ...
Unpacking php8.1-mysql (8.1.2-1ubuntu2.14) ...
Selecting previously unselected package php-mysql.
Preparing to unpack .../10-php-mysql_2:8.1+92ubuntu1_all.deb ...
Unpacking php-mysql (2:8.1+92ubuntu1) ...
Setting up php-common (2:92ubuntu1) ...
```

## Step3: Install MySQL database

### sudo apt install mysql-server

```
ubuntu@ip-172-31-47-219:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libbgi-fast-perl libbgi-pm-perl libclone-perl libencode-locale-perl
  libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0db1
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 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 libbusiness-isbn-perl libwww-perl
  perl-tidy
The following NEW packages will be installed:
  libbgi-fast-perl libbgi-pm-perl libclone-perl libencode-locale-perl
  libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi0db1
  libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 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 65 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] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.34-0ubuntu22.04.1 [2754 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-8.0 amd64 8.0.34-0ubuntu22.04.1 [22.7 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-3build1 [7462 B]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libmecab2 amd64 0.998-4build1 [199 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-lite23 amd64 3.12.4-1ubuntu7.22.04.1 [209 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-core-8.0 amd64 8.0.34-0ubuntu22.04.1 [17.5 MB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-8.0 amd64 8.0.34-0ubuntu22.04.1 [1437 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-tagset-perl all 3.28-4 [12.5 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liburi-perl all 5.18-1 [78.8 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-parser-perl amd64 3.76-1build2 [88.4 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libbgi-pm-perl all 4.54-1 [188 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi0db1 amd64 2.4.2-2build2 [28.0 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-perl amd64 0.83-4-1build1 [21.0 kB]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libbgi-fast-perl all 3.12.15-1 [10.5 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libclone-perl amd64 0.45-1build3 [11.0 kB]
Get:17 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libencode-locale-perl all 3.80-1.1 [51.0 kB]
Get:18 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libfcgi-bin amd64 2.4.2-2build2 [11.2 kB]
Get:19 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-template-perl all 2.97-1.1 [59.1 kB]
Get:20 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libtimedate-perl all 2.3300-2 [34.0 kB]
Get:21 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhttp-date-perl all 6.85-1 [9928 B]
Get:22 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libio-html-perl all 1.004-2 [15.4 kB]
Get:23 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblwp-mediatypes-perl all 6.04-1 [19.5 kB]
Get:24 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhttp-message-perl all 6.36-1 [76.8 kB]
Get:25 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mecab-utils amd64 0.998-4build1 [4859 B]
Get:26 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mecab-ipadic all 2.7.0-20070808+main-3 [6718 kB]
Get:27 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mecab-ipadic-utf8 all 2.7.0-20070808+main-3 [4304 B]
Get:28 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-8.0 all 8.0.34-0ubuntu22.04.1 [9468 B]
Fetched 29.6 MB in 1s (38.6 MB/s)
Preconfiguring packages ...
Selecting previously unselected package mysql-common.
(Reading database ... 98815 files and directories currently installed.)
Preparing to unpack .../8-mysql-common_5.8+1.0.8_all.deb ...
Unpacking mysql-common (5.8+1.0.8) ...
Selecting previously unselected package mysql-client-core-8.0.
```

### Configuration to mysql server.

Change the mysql authentication plugin to mysql\_native\_password. So that we can login to mysql-server with normal password.

### First login to MySQL Server

### sudo mysql -u root

### Change the authentication plugin

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password
by 'TestPassword@123';
```

Create a new user other than root account in order to use the wordpress.

```
CREATE USER 'wp_user'@localhost IDENTIFIED BY
'Testpassword@123';
```

Create a new database for wordpress



```
CREATE DATABASE wp;
```

Grant all the privileges to the database 'wp' to the 'wp\_user'.

```
GRANT ALL PRIVILEGES ON wp.* TO 'wp_user'@localhost;
```

We configured the mysql-server

```
[ubuntu@ip-172-31-47-219:~]$ sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.34-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, 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 'Testpassword@123';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE USER 'wp_user'@localhost IDENTIFIED BY 'Testpassword@123';
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE DATABASE wp;
Query OK, 1 row affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON wp.* TO 'wp_user'@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> ^DBye
[ubuntu@ip-172-31-47-219:~]$
```

## **Step 4: Download and install wordpress.**

Goto /tmp directory

Download the latest.tar.gz

```
cd /tmp
```

```
wget https://wordpress.org/latest.tar.gz
```

```
unzip
```

```
tar -xvf latest.tar.gz
```

Move the wordpress folder to the apache document root

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

```

ubuntu@ip-172-31-47-219:/$ cd /tmp/
ubuntu@ip-172-31-47-219:/tmp$ wget https://wordpress.org/latest.tar.gz
--2023-10-04 05:39:26-- 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: 23447259 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

latest.tar.gz      100%[=====] 22.36M  48.3MB/s   in 0.5s

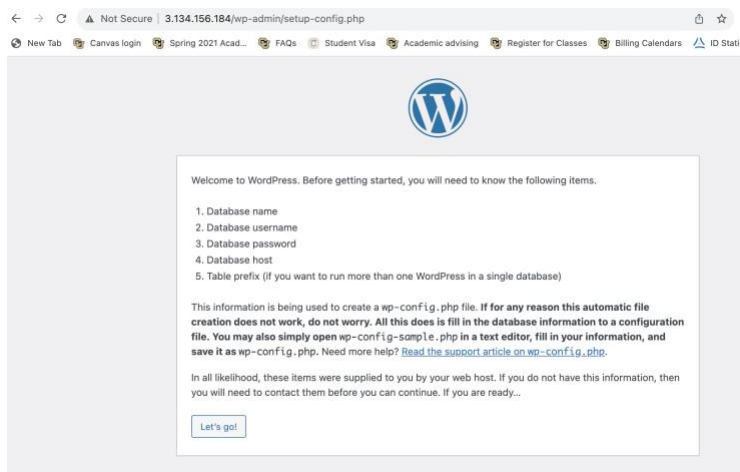
2023-10-04 05:39:26 (48.3 MB/s) - 'latest.tar.gz' saved [23447259/23447259]

ubuntu@ip-172-31-47-219:/tmp$ ls
latest.tar.gz
snap-private-tmp
systemd-private-7aa8dd189256437692b9f4dd30cae343-apache2.service-5jWFuc
systemd-private-7aa8dd189256437692b9f4dd30cae343-chrony.service-nG6KiU
systemd-private-7aa8dd189256437692b9f4dd30cae343-systemd-logind.service-AtFOWu
systemd-private-7aa8dd189256437692b9f4dd30cae343-systemd-resolved.service-fJyEu
tmp.oaaQbP4Qk
ubuntu@ip-172-31-47-219:/tmp$ tar -xvf latest.tar.gz
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
wordpress/wp-login.php
wordpress/wp-settings.php
wordpress/license.txt
wordpress/wp-content/
wordpress/wp-content/themes/
wordpress/wp-content/themes/twentytwentythree/
wordpress/wp-content/themes/twentytwentythree/theme.json
wordpress/wp-content/themes/twentytwentythree/parts/
wordpress/wp-content/themes/twentytwentythree/parts/footer.html
wordpress/wp-content/themes/twentytwentythree/parts/comments.html
wordpress/wp-content/themes/twentytwentythree/parts/header.html
wordpress/wp-content/themes/twentytwentythree/parts/post-meta.html
wordpress/wp-content/themes/twentytwentythree/patterns/
wordpress/wp-content/themes/twentytwentythree/patterns/hidden-404.php
wordpress/wp-content/themes/twentytwentythree/patterns/post-meta.php
wordpress/wp-content/themes/twentytwentythree/patterns/hidden-no-results.php
wordpress/wp-content/themes/twentytwentythree/patterns/call-to-action.php
wordpress/wp-content/themes/twentytwentythree/patterns/footer-default.php

ubuntu@ip-172-31-47-219:/tmp$ sudo mv wordpress/ /var/www/html/
ubuntu@ip-172-31-47-219:/tmp$ ls
latest.tar.gz
snap-private-tmp
systemd-private-7aa8dd189256437692b9f4dd30cae343-apache2.service-5jWFuc
systemd-private-7aa8dd189256437692b9f4dd30cae343-chrony.service-nG6KiU
systemd-private-7aa8dd189256437692b9f4dd30cae343-systemd-logind.service-AtFOWu
systemd-private-7aa8dd189256437692b9f4dd30cae343-systemd-resolved.service-fJyEu
tmp.oaaQbP4Qk
ubuntu@ip-172-31-47-219:/tmp$ cd /var/www/html/
ubuntu@ip-172-31-47-219:/var/www/html$ ls
index.html  wordpress

```

When you open the public IPV4 address in the browser(<http://3.134.156.184/wordpress>), then the wordpress installation page is displayed.



Install the wordpress, click on “Let’s go!”

Next, configure the database.

Database Name: wp

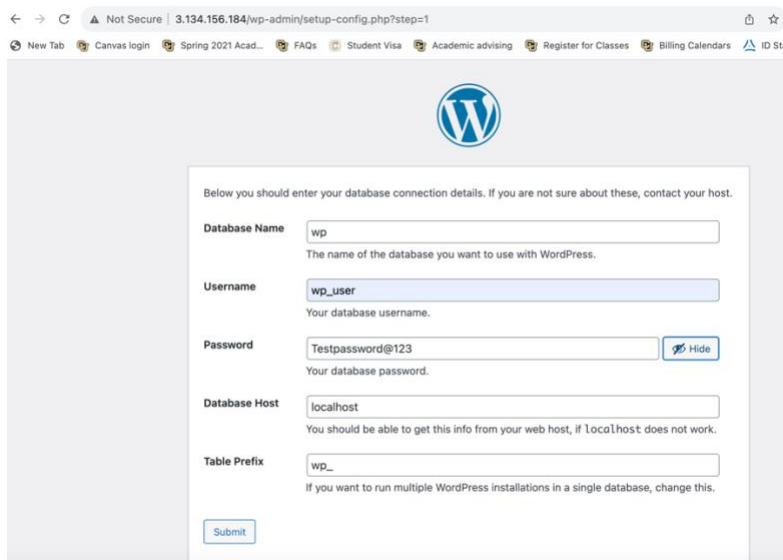
Username: wp\_user

Password: Testpassword@123

Database Host: localhost

Table Prefix: wp\_

Click on “submit”



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**  [Hide](#)  
Your database password.

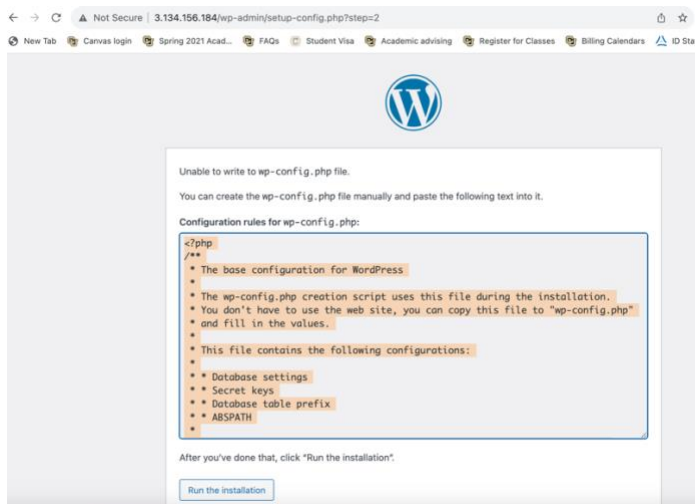
**Database Host**   
You should be able to get this info from your web host, if localhost does not work.

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

[Submit](#)

There is an error “Unable to write wp-config.php file”

So, copy the php code



Unable to write to wp-config.php file.

You can create the wp-config.php file manually and paste the following text into it.

Configuration rules for wp-config.php:

```
<?php
/**
 * The base configuration for WordPress.
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 */
```

After you've done that, click "Run the installation".

[Run the installation](#)

Goto terminal, Goto /var/www/html/wordpress/

```
cd /var/www/html/wordpress/
```

In the path, create a new file wp-config.php

```
nano wp-config.php
```

paste the php code.

```
ubuntu@ip-172-31-47-219:/$ cd /var/www/html/wordpress/  
ubuntu@ip-172-31-47-219:/var/www/html/wordpress$ nano wp-config.php  
ubuntu@ip-172-31-47-219:/var/www/html/wordpress$ █
```

Click on “Run the installation” in the browser.

Fill the information needed

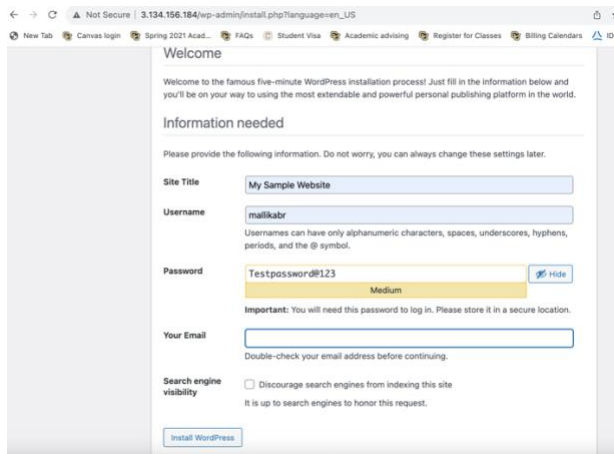
Site Title: my sample website

Username: mallikabr

Password: Testpassword@123

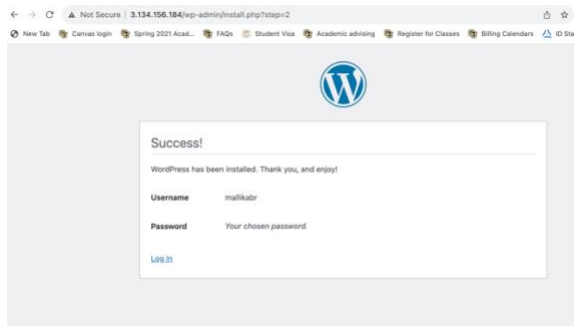
Your email

Click on “Install WordPress”

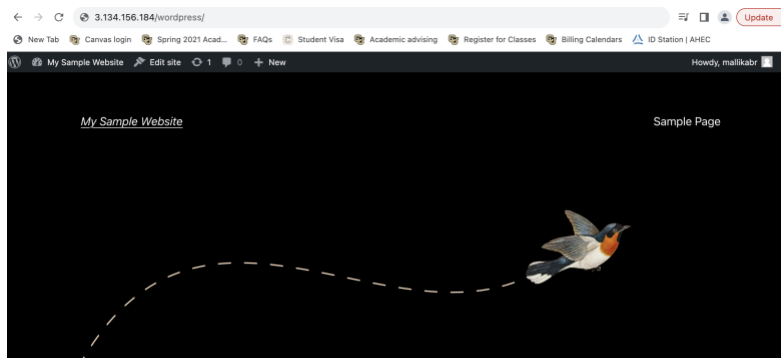


The screenshot shows the WordPress installation 'Welcome' screen in a web browser. The page title is 'Welcome' and the URL is '3.134.156.184/wp-admin/install.php?language=en\_US'. The page contains a 'Welcome' message, followed by an 'Information needed' section. It asks for 'Site Title' (filled with 'My Sample Website'), 'Username' (filled with 'mallikabr'), 'Password' (filled with 'Testpassword@123', showing a strength indicator of 'Medium'), and 'Your Email'. There is also a checkbox for 'Search engine visibility' which is unchecked. At the bottom, there is an 'Install WordPress' button.

WordPress is now installed.



Open <http://3.134.156.184/wordpress/>



We don't have to go to the sub path i.e., /wordpress (<http://3.134.156.184/wordpress/>)

The wordpress website has to be served on the root directory (<http://3.134.156.184/>)

For that goto terminal, goto /etc/apache2/sites-available/

```
cd /etc/apache2/sites-available/
```

Then open the 000-default.conf (configuration file)

```
sudo nano 000-default.conf
```

change the DocumentRoot from /var/www/html to /var/www/html/wordpress

```
GNU nano 6.2 000-default.conf
VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/wordpress

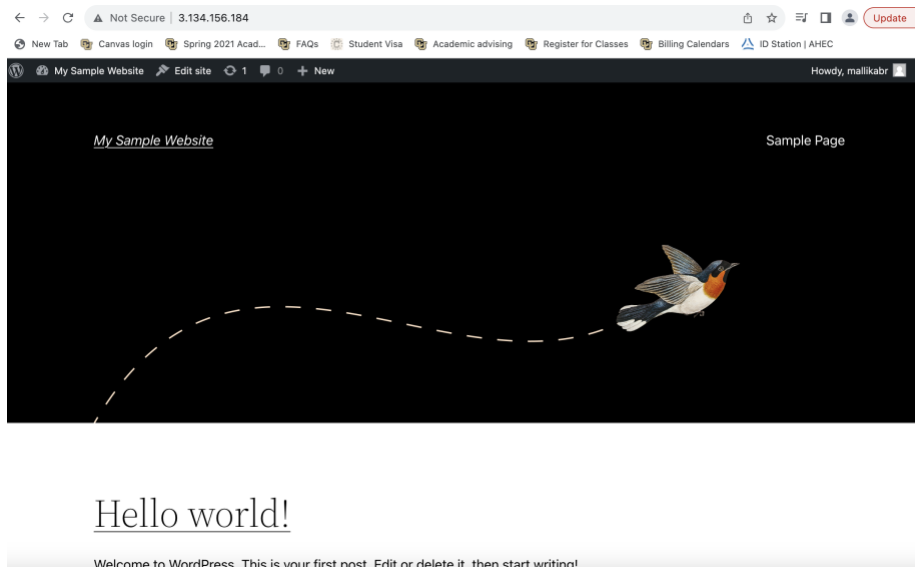
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

To restart the apache2

**sudo systemctl restart apache2**

```
[ubuntu@ip-172-31-47-219:/var/www/html/wordpress$ cd /etc/apache2/sites-available/
ubuntu@ip-172-31-47-219:/etc/apache2/sites-available$ ls
ubuntu@ip-172-31-47-219:/etc/apache2/sites-available$ sudo systemctl restart apache2
ubuntu@ip-172-31-47-219:/etc/apache2/sites-available$
```

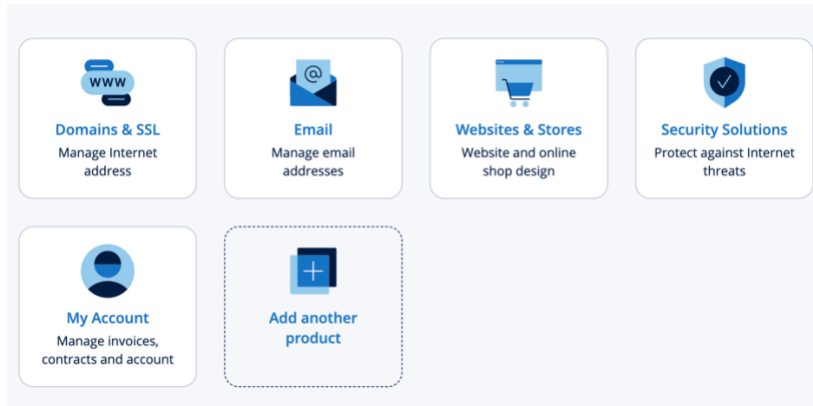
This worked!!



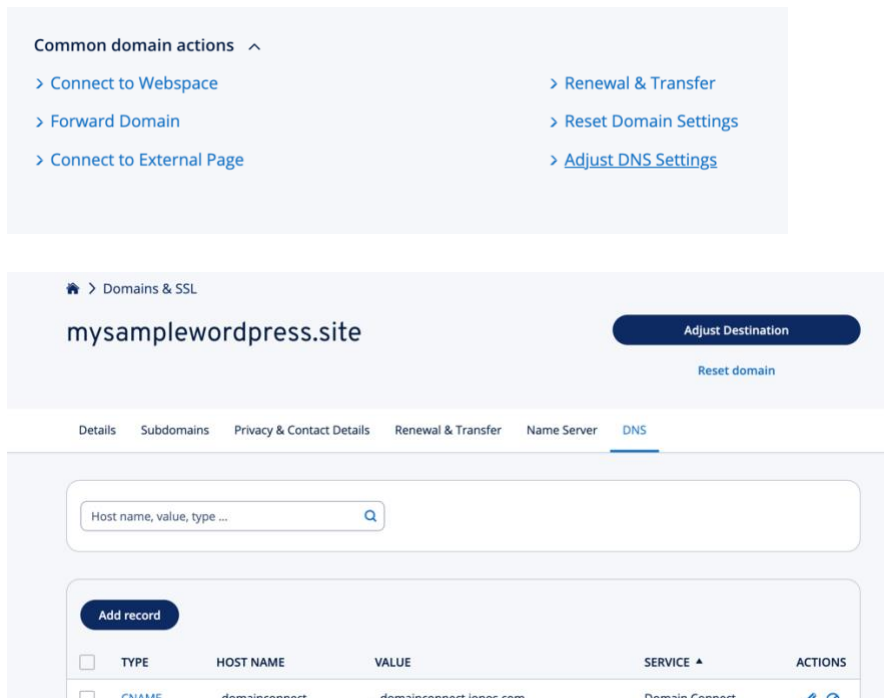
## **Step5: Link a custom domain to a website**

My domain Name: mysamplewordpress.site from IONOS. Now link this domain to the wordpress.

Login to IONOS portal and click on “Domains & SSL” in home page.



Click on “Adjust DNS Settings”, then click on “Add record”.



Choose A type and points to public IPV4 address i.e., 3.134.156.184 and TTL set to 60 and click on “save”.

## Add a DNS record

Type [A](#)

Host Name

Points to

[TTL](#)

Preview mysamplewordpress.site 60 IN A  
3.134.156.184

Preview for www www.mysamplewordpress.site 60 IN A  
3.134.156.184  
[> Do not add DNS record for www](#)

And there are some additional changes that needs to be modified in the configuration file

```
cd /etc/apache2/sites-available
```

```
sudo nano 000-default.conf
```

add the below two lines in the .conf file

```
ServerName mysamplewordpress.site
```

```
ServerAlias www.mysamplewordpress.site
```



```

GNU nano 6.2 000-default.conf
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/wordpress

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ServerName mysamplewordpress.site
ServerAlias www.mysamplewordpress.site

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

```

Restart the apache2

**sudo systemctl restart apache2**

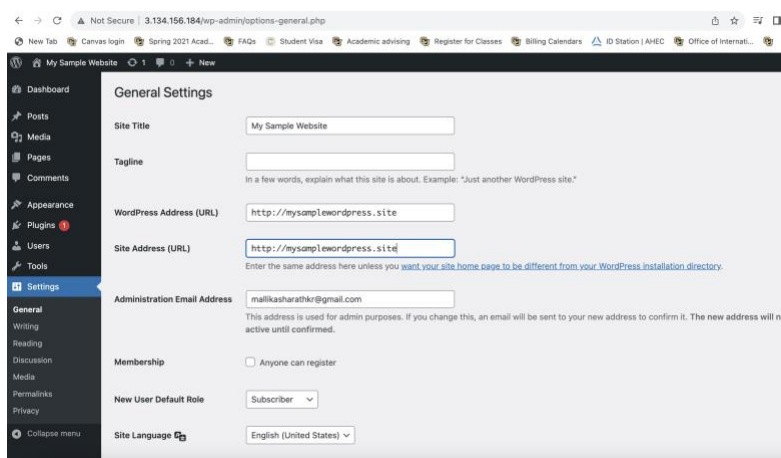
Finally, domain name is pointing to the AWS instance. Now we can access the wordpress using the domain name.

Visit the website.

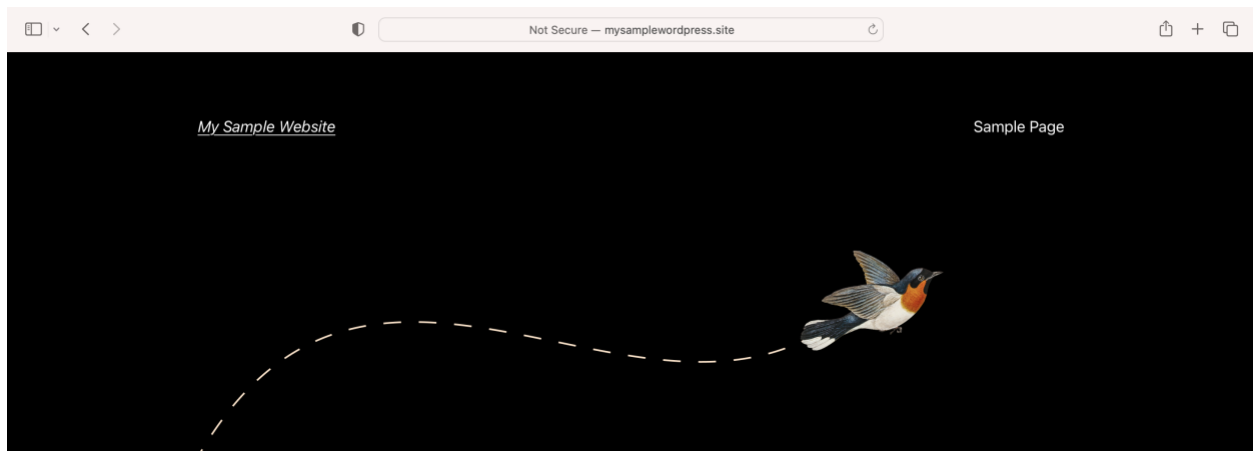
[www.mysamplewordpress.site](http://www.mysamplewordpress.site)

Now, when you visit [www.mysamplewordpress.site/wp-admin](http://www.mysamplewordpress.site/wp-admin), it is redirecting the public IPV4 address. To fix this,

Login to the wordpress, In the Dashboard, Goto Settings, then click on General. Modify the Wordpress Address (URL): <http://mysamplewordpress.site> Site Address(URL): <http://mysamplewordpress.site> Click on “Save changes”



Now, the website points to domain name.



Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

October 4, 2023

Here the website is not secure. It has to work over https.  
We need to install certbot.

**sudo apt install certbot python3-certbot-apache**

```
[ubuntu@ip-172-31-47-219:~]$ cd /etc/apache2/sites-available/
[ubuntu@ip-172-31-47-219:/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-requests-toolbelt python3-rfc3339 python3-zope.component
  python3-zope.event python3-zope.hookable
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-requests-toolbelt python3-rfc3339
  python3-zope.component python3-zope.event python3-zope.hookable
0 upgraded, 16 newly installed, 0 to remove and 60 not upgraded.
Need to get 1552 kB of archives.
After this operation, 7681 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Augeas-lenses all 1.13.0-1 [321 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 libaugeas0 amd64 1.13.0-1 [200 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 python3-josepy all 1.10.0-1 [22.0 kB]
```

Run the command to secure the apache with SSL certificate  
**sudo certbot --apache**

prompts for an email address and enter

```

ubuntu@ip-172-31-47-219:/etc/apache2/sites-available$ sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log

Which names would you like to activate HTTPS for?
- - - - -
1: mysamplewordpress.site
2: www.mysamplewordpress.site
- - - - -
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Requesting a certificate for mysamplewordpress.site and www.mysamplewordpress.site

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/mysamplewordpress.site/fullchain.pem
Key is saved at: /etc/letsencrypt/live/mysamplewordpress.site/privkey.pem
This certificate expires on 2024-01-03.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for mysamplewordpress.site to /etc/apache2/sites-available/000-default-le-ssl.conf
Successfully deployed certificate for www.mysamplewordpress.site to /etc/apache2/sites-available/000-default-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://mysamplewordpress.site and https://www.mysamplewordpress.site

- - - - -
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
* Donating to EFF: https://eff.org/donate-le
- - - - -
ubuntu@ip-172-31-47-219:/etc/apache2/sites-available$

```

Successfully deployed certificate for both `mysamplewordpress.site` and `www.mysamplewordpress.site`

The certificate issued by R3, Let's Encrypt

Certificate Viewer: mysamplewordpress.site

General	Details
<b>Issued To</b> Common Name (CN) mysamplewordpress.site Organization (O) <Not Part Of Certificate> Organizational Unit (OU) <Not Part Of Certificate>	
<b>Issued By</b> Common Name (CN) R3 Organization (O) Let's Encrypt Organizational Unit (OU) <Not Part Of Certificate>	
<b>Validity Period</b>	

Now the domain is secure and works over https. So, you can view wordpress website on the custom domain.

