

Automated WordPress Deployment using Ansible on AWS EC2 (Linux)

Cloud & DevOps Automation Project



Prepared by

Nandu Sivadas

Cloud & DevOps Enthusiast

Project Objective

To automate the deployment of a secure WordPress hosting environment on AWS EC2 using Ansible Roles. The project implements Infrastructure as Code (IaC) to automatically configure Nginx, PHP, MariaDB, WordPress, phpMyAdmin, SFTP, and SSL, ensuring a consistent and repeatable deployment process.

Project Overview

This project automates the deployment of a secure WordPress hosting environment on an AWS EC2 Linux server using Ansible Roles. It provisions and configures Nginx, PHP, MariaDB, WordPress, phpMyAdmin, SFTP, and SSL by following Infrastructure as Code (IaC) principles, ensuring a secure, scalable, and repeatable deployment process.

Objectives

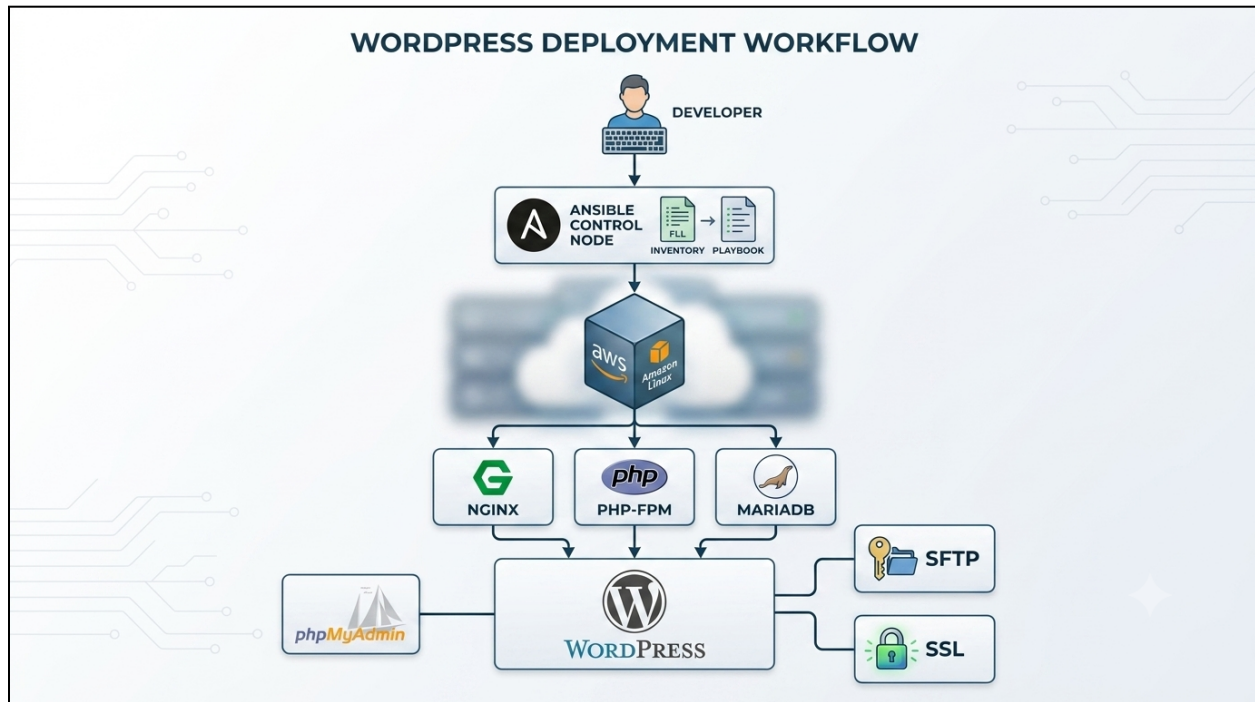
The primary objectives of this project are to:

- Automate WordPress deployment using Ansible Roles.
- Deploy and configure the LEMP stack (Nginx, PHP, and MariaDB).
- Create dedicated website users and directory structures.
- Configure WordPress and phpMyAdmin.
- Enable secure SFTP access for website management.
- Secure the website with Let's Encrypt SSL certificates.
- Apply Infrastructure as Code (IaC) principles for automated deployment.
- Ensure a consistent, secure, and repeatable hosting environment.

Key Features

- Automated WordPress deployment using Ansible Roles
- Infrastructure as Code (IaC) implementation
- Nginx, PHP, and MariaDB configuration
- Secure SFTP access
- phpMyAdmin integration
- HTTPS using Let's Encrypt SSL
- Modular role-based automation
- End-to-end deployment verification

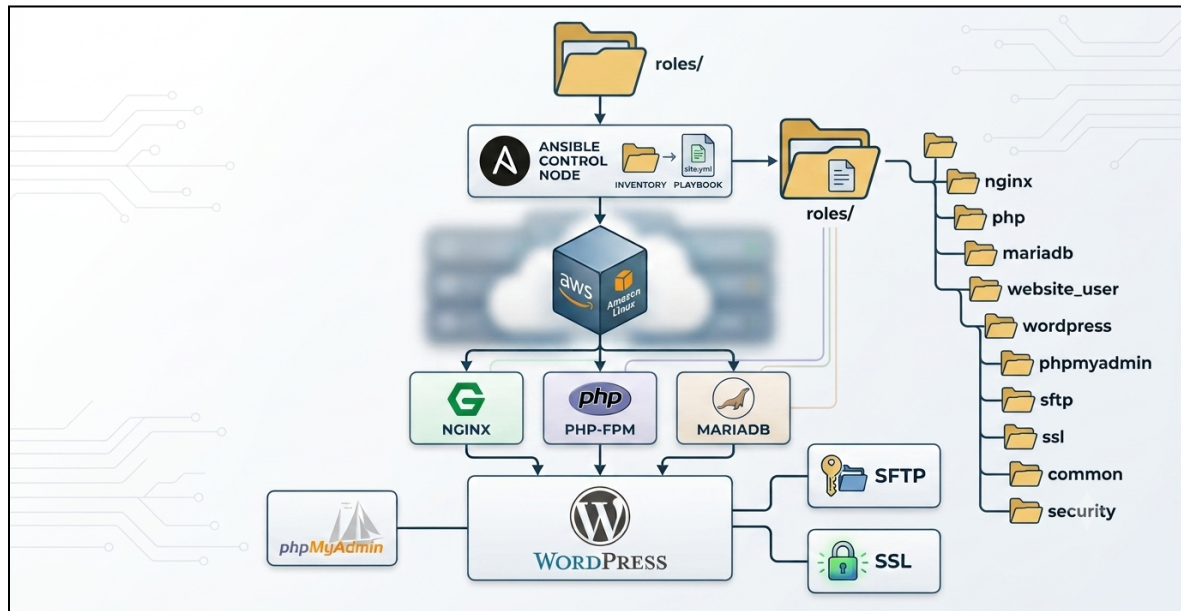
Architecture Diagram



Technologies Used

- AWS EC2
- Ansible
- Ansible Roles
- Nginx
- PHP 8+
- MariaDB
- WordPress
- phpMyAdmin
- OpenSSH (SFTP)
- Certbot
- Let's Encrypt SSL
- Git & **Github**

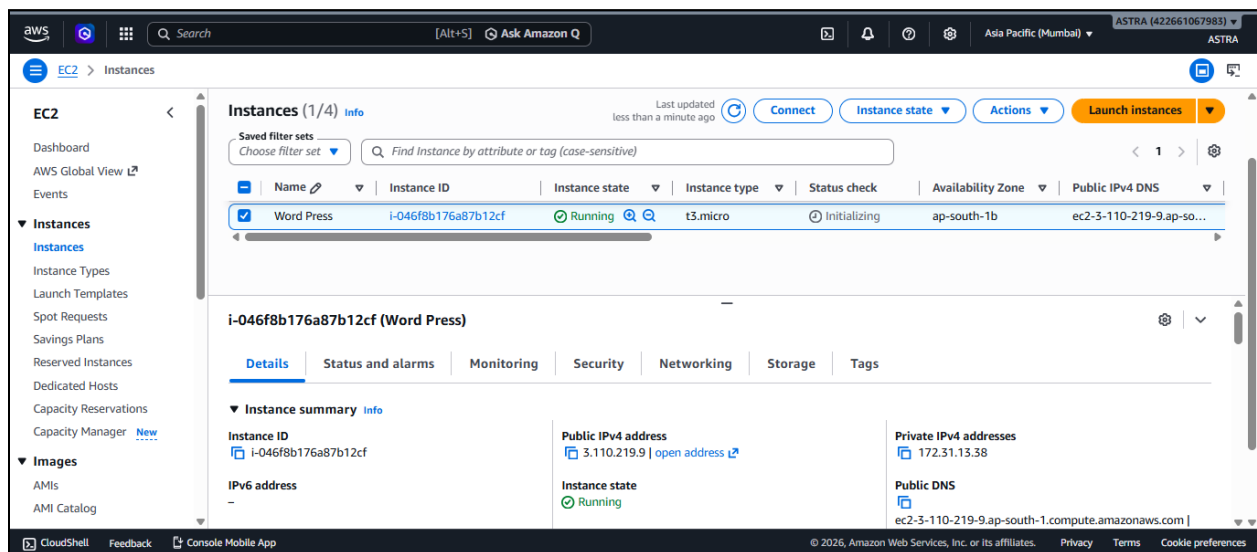
Project Structure



Task 01 - AWS Infrastructure Setup

Launch AWS EC2Instance

- Login to **AWS** Console and Open **EC2** Dashboard.
- Launch Amazon Linux 2023 instance.
- Select t3.micro.
- Create or choose an existing key pair.
- Configure Security Group: **SSH-22, HTTP-80, HTTPS-443**.



Connect to EC2 Instance using Putty

- Login as ec2-user

```
ec2-user@ip-172-31-13-38:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
#  
~\_##### Amazon Linux 2023  
~~~\_#####\  
~~~\_###|  
~~~\_#/ https://aws.amazon.com/linux/amazon-linux-2023  
~~~V~'-'->  
~~~~  
~~~.-.  
~~~/_m/'-'  
[ec2-user@ip-172-31-13-38 ~]$  
[ec2-user@ip-172-31-13-38 ~]$
```

Task 02 - Ansible Installation

(i) Install Ansible

```
# yum install ansible -y
```

```

ec2-user@ip-172-31-13-38: ~$ sudo dnf update -y
Amazon Linux 2023 Kernel Livepatch repository                                443 kB/s | 55 kB      00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-13-38 ~]$ sudo dnf install ansible -y
Last metadata expiration check: 0:00:41 ago on Tue Jun 30 14:47:57 2026.
Dependencies resolved.

=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
ansible                               noarch             8.3.0-1.amzn2023.0.2  amazonlinux         32 M
Installing dependencies:
ansible-core                          x86_64            2.15.3-1.amzn2023.0.12 amazonlinux         2.5 M
git-core                             x86_64            2.50.1-1.amzn2023.0.1  amazonlinux         4.9 M
sshpass                              x86_64            1.09-6.amzn2023.0.1    amazonlinux         28 k
=====

Transaction Summary
-----
Install 4 Packages

Total download size: 40 M
Installed size: 552 M
Downloading Packages:
(1/4): git-core-2.50.1-1.amzn2023.0.1.x86_64.rpm                    51 MB/s | 4.9 MB      00:00
(2/4): ansible-core-2.15.3-1.amzn2023.0.12.x86_64.rpm              20 MB/s | 2.5 MB      00:00
(3/4): sshpass-1.09-6.amzn2023.0.1.x86_64.rpm                      974 kB/s | 28 kB      00:00
(4/4): ansible-8.3.0-1.amzn2023.0.2.noarch.rpm                     66 MB/s | 32 MB      00:00
-----
Total                                                                    74 MB/s | 40 MB      00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing :
  Installing : sshpass-1.09-6.amzn2023.0.1.x86_64

```

(ii) Verify Installation

ansible --version

```
[ec2-user@ip-172-31-13-38 ~]$ ansible --version
ansible [core 2.15.3]
  config file = None
  configured module search path = ['/home/ec2-user/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3.9/site-packages/ansible
  ansible collection location = /home/ec2-user/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.9.25 (main, May 25 2026, 00:00:00) [GCC 11.5.0 20240719 (Red Hat 11.5.0-5)] (/usr/bin/python3.9)
  jinja version = 3.1.4
  libyaml = True
[ec2-user@ip-172-31-13-38 ~]$
```

Task 03 - Project Initialization

(i) Create Project Directory

```
[ec2-user@ip-172-31-13-38 ~]$ mkdir wordpress-project
[ec2-user@ip-172-31-13-38 ~]$ cd wordpress-project
[ec2-user@ip-172-31-13-38 wordpress-project]$ pwd
/home/ec2-user/wordpress-project
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

(ii) Create inventory file

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim inventory
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat inventory
[web]
localhost ansible_connection=local
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

(iii) Create Main Playbook

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim site.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat site.yml
---
- hosts: web
  become: yes

  roles:
    - nginx
    - php
    - mariadb
    - wordpress
    - phpmyadmin
    - sftp
    - ssl
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

Task 04 - Ansible Role Creation

Initialize Ansible Role

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init nginx
- Role nginx was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init php
- Role php was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init mariadb
- Role mariadb was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init wordpress
- Role wordpress was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init phpmyadmin
- Role phpmyadmin was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init sftp
- Role sftp was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-galaxy init ssl
- Role ssl was created successfully
[ec2-user@ip-172-31-13-38 wordpress-project]$ ls
inventory  mariadb  nginx  php  phpmyadmin  sftp  site.yml  ssl  wordpress
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

Step 01: Nginx Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim nginx/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat nginx/tasks/main.yml
---
# tasks file for nginx
- name: Install Nginx
  dnf:
    name: nginx
    state: present

- name: Start and Enable Nginx
  service:
    name: nginx
    state: started
    enabled: yes
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Execute Nginx Role

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-playbook -i inventory site.yml

PLAY [web] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]

TASK [nginx : Install Nginx] *****
changed: [localhost]

TASK [nginx : Start and Enable Nginx] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Nginx service

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ nginx -v
nginx version: nginx/1.30.2
[ec2-user@ip-172-31-13-38 wordpress-project]$ systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Tue 2026-06-30 14:58:57 UTC; 59s ago
     Process: 26828 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 26829 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 26830 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 26831 (nginx)
      Tasks: 3 (limit: 1059)
     Memory: 3.4M
        CPU: 60ms
    CGroup: /system.slice/nginx.service
            └─26831 "nginx: master process /usr/sbin/nginx"
              └─26832 "nginx: worker process"
                └─26833 "nginx: worker process"

Jun 30 14:58:57 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server:
Jun 30 14:58:57 ip-172-31-13-38.ap-south-1.compute.internal nginx[26829]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jun 30 14:58:57 ip-172-31-13-38.ap-south-1.compute.internal nginx[26829]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jun 30 14:58:57 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server:
lines 1-19/19 (END)
```

Step 02: Configure Nginx Virtual Host

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo vim /etc/nginx/conf.d/wordpress.conf
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat /etc/nginx/conf.d/wordpress.conf
server {
    listen 80;
    server_name _;

    root /home/admin/personalblog/public;
    index index.php index.html;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_pass unix:/run/php-fpm/www.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Nginx Configuration and Restart Nginx service

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo nginx -t
nginx: [warn] conflicting server name "_" on 0.0.0.0:80, ignored
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Virtual Host Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Tue 2026-06-30 15:02:31 UTC; 22s ago
     Process: 27011 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 27012 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 27013 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 27014 (nginx)
      Tasks: 3 (limit: 1059)
     Memory: 3.7M
        CPU: 59ms
    CGroup: /system.slice/nginx.service
            └─27014 "nginx: master process /usr/sbin/nginx"
              └─27015 "nginx: worker process"
                └─27016 "nginx: worker process"

Jun 30 15:02:31 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server:
Jun 30 15:02:31 ip-172-31-13-38.ap-south-1.compute.internal nginx[27012]: nginx: [warn] conflicting server name "_" on 0.0.0.0:80, ignored
Jun 30 15:02:31 ip-172-31-13-38.ap-south-1.compute.internal nginx[27012]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jun 30 15:02:31 ip-172-31-13-38.ap-south-1.compute.internal nginx[27012]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jun 30 15:02:31 ip-172-31-13-38.ap-south-1.compute.internal nginx[27013]: nginx: [warn] conflicting server name "_" on 0.0.0.0:80, ignored
Jun 30 15:02:31 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server:
lines 1-21/21 (END)
```

Step 03: Php Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim php/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat php/tasks/main.yml
---
# tasks file for php
- name: Install PHP
  dnf:
    name:
      - php
      - php-fpm
      - php-mysqlnd
      - php-gd
      - php-mbstring
      - php-xml
    state: present

- name: Start PHP-FPM
  service:
    name: php-fpm
    state: started
    enabled: yes
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Execute Php Role

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-playbook -i inventory site.yml

PLAY [web] *********************************************************************

TASK [Gathering Facts] *********************************************************
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python3.9, but future installation
of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]

TASK [nginx : Install Nginx] ***************************************************
ok: [localhost]

TASK [nginx : Start and Enable Nginx] *******************************************
ok: [localhost]

TASK [php : Install PHP] ********************************************************
changed: [localhost]

TASK [php : Start PHP-FPM] *****************************************************
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                : ok=5    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Php service

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ php -v
PHP 8.5.7 (cli) (built: Jun  2 2026 20:59:56) (NTS gcc x86_64)
Copyright (c) The PHP Group
Built by Amazon Linux
Zend Engine v4.5.7, Copyright (c) Zend Technologies
    with Zend OPcache v8.5.7, Copyright (c), by Zend Technologies
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl status php-fpm
● php-fpm.service - The PHP FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php-fpm.service; enabled; preset: disabled)
   Active: active (running) since Tue 2026-06-30 15:07:11 UTC; 56s ago
     Main PID: 27777 (php-fpm)
    Status: "Processes active: 0, idle: 5, Requests: 0, slow: 0, Traffic: 0.00req/sec"
       Tasks: 6 (limit: 1059)
      Memory: 12.0M
         CPU: 60ms
    CGroup: /system.slice/php-fpm.service
            └─27777 "php-fpm: master process (/etc/php-fpm.conf)"
              └─27778 "php-fpm: pool www"
                └─27779 "php-fpm: pool www"
                  └─27780 "php-fpm: pool www"
                    └─27781 "php-fpm: pool www"
                      └─27782 "php-fpm: pool www"

Jun 30 15:07:10 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Starting php-fpm.service - The PHP FastCGI Process Manager...
Jun 30 15:07:11 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Started php-fpm.service - The PHP FastCGI Process Manager.
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

Step 04: Mariadb Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim mariadb/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat mariadb/tasks/main.yml
---
# tasks file for mariadb
- name: Install MariaDB Server
  dnf:
    name:
      - mariadb105-server
      - mariadb105
    state: present

- name: Install pip
  dnf:
    name: python3-pip
    state: present

- name: Install PyMySQL
  pip:
    name: PyMySQL

- name: Start and Enable MariaDB
  service:
    name: mariadb
    state: started
    enabled: yes
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Execute Mariadb Role

```
ec2-user@ip-172-31-13-38:~/wordpress-project
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-playbook -i inventory site.yml

PLAY [web] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]

TASK [nginx : Install Nginx] *****
ok: [localhost]

TASK [nginx : Start and Enable Nginx] *****
ok: [localhost]

TASK [php : Install PHP] *****
ok: [localhost]

TASK [php : Start PHP-FPM] *****
ok: [localhost]

TASK [mariadb : Install MariaDB Server] *****
changed: [localhost]

TASK [mariadb : Install pip] *****
changed: [localhost]

TASK [mariadb : Install PyMySQL] *****
changed: [localhost]

TASK [mariadb : Start and Enable MariaDB] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=9    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Mariadb service

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ mariadb --version
mariadb Ver 15.1 Distrib 10.5.29-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ mariadb --version
mariadb Ver 15.1 Distrib 10.5.29-MariaDB, for Linux (x86_64) using EditLine wrapper
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: active (running) since Tue 2026-06-30 15:12:29 UTC; 2min 31s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 29862 ExecStartPre=/usr/libexec/mariadb-check-socket (code=exited, status=0/SUCCESS)
   Process: 29884 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir mariadb.service (code=exited, status=0/SUCCESS)
   Process: 29986 ExecStartPost=/usr/libexec/mariadb-check-upgrade (code=exited, status=0/SUCCESS)
  Main PID: 29968 (mariadbd)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 1059)
    Memory: 66.4M
       CPU: 568ms
    CGroup: /system.slice/mariadb.service
            └─29968 /usr/libexec/mariadbd --basedir=/usr

Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: The second is mysql@localhost, it has no p
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: you need to be the system 'mysql' user to
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: After connecting you can set the password,
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: able to connect as any of these users with
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: See the MariaDB Knowledgebase at https://m
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: Please report any problems at https://mari
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: The latest information about MariaDB is av
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: Consider joining MariaDB's strong and vibr
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal mariadb-prepare-db-dir[29923]: https://mariadb.org/get-involved/
Jun 30 15:12:29 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.
lines 1-26/26 (END)
```

→ Verify PyMysql Installation

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ pip3 show PyMySQL
Name: PyMySQL
Version: 1.2.0
Summary: Pure Python MySQL Driver
Home-page:
Author:
Author-email: Inada Naoki <songofacandy@gmail.com>, Yutaka Matsubara <yutaka.matsubara@gmail.com>
License:
Location: /usr/local/lib/python3.9/site-packages
Requires:
Required-by:
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

Step 05: WordPress Role Configuration

- Website User Management
- Website Directory Setup
- Database Configuration

```
ec2-user@ip-172-31-13-38:~/wordpress-project
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim wordpress/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat wordpress/tasks/main.yml
---
# tasks file for wordpress

# Website User Management
- name: Create Website User
  user:
    name: admin
    shell: /bin/bash
    create_home: yes
    state: present

# Website Directory Setup
- name: Create Website Directory
  file:
    path: /home/admin/personalblog/public
    state: directory
    owner: admin
    group: admin
    mode: '0755'
    recurse: yes

# Database Configuration
- name: Create WordPress Database
  mysql_db:
    name: wordpress
    state: present
    login_unix_socket: /var/lib/mysql/mysql.sock

- name: Create WordPress Database User
  mysql_user:
    name: wpuser
    password: wordpress@123
    priv: "wordpress.*:ALL"
    state: present
    login_unix_socket: /var/lib/mysql/mysql.sock
```

- Wordpress Installation and Configuration.
- Security and Permissions.

```
# Wordpress Installation
- name: Download Wordpress
  get_url:
    url: https://wordpress.org/latest.tar.gz
    dest: /tmp/wordpress.tar.gz

- name: Extract Wordpress
  unarchive:
    src: /tmp/wordpress.tar.gz
    dest: /tmp/
    remote_src: yes

- name: Copy Wordpress Files
  copy:
    src: /tmp/wordpress/
    dest: /home/admin/personalblog/public/
    remote_src: yes

# Wordpress Configuration
- name: Configure wp-config.php
  template:
    src: wp-config.php.j2
    dest: /home/admin/personalblog/public/wp-config.php
    owner: admin
    group: admin
    mode: '0644'

# Security and Permissions
- name: Set Permissions
  file:
    path: /home/admin/personalblog
    owner: admin
    group: admin
    recurse: yes
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

➔ Execute Wordpress Role

```
TASK [wordpress : Create Website User] *****
ok: [localhost]

TASK [wordpress : Create Website Directory] *****
changed: [localhost]

TASK [wordpress : Create WordPress Database] *****
ok: [localhost]

TASK [wordpress : Create WordPress Database User] *****
ok: [localhost]

TASK [wordpress : Download Wordpress] *****
ok: [localhost]

TASK [wordpress : Extract Wordpress] *****
ok: [localhost]

TASK [wordpress : Copy Wordpress Files] *****
ok: [localhost]

TASK [wordpress : Configure wp-config.php] *****
changed: [localhost]

TASK [wordpress : Set Permissions] *****
ok: [localhost]

PLAY RECAP *****
localhost : ok=19  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

➔ Verify Wordpress Installation

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo ls -la /home/admin/personalblog/public
total 264
drwxr-xr-x. 5 admin admin 16384 Jun 30 15:24 .
drwxr-xr-x. 3 admin admin 20 Jun 30 15:21 ..
-rwxr-xr-x. 1 admin admin 405 Jun 30 15:21 index.php
-rwxr-xr-x. 1 admin admin 19903 Jun 30 15:21 license.txt
-rwxr-xr-x. 1 admin admin 7406 Jun 30 15:21 readme.html
-rwxr-xr-x. 1 admin admin 7371 Jun 30 15:21 wp-activate.php
drwxr-xr-x. 9 admin admin 16384 May 19 17:16 wp-admin
-rwxr-xr-x. 1 admin admin 351 Jun 30 15:21 wp-blog-header.php
-rwxr-xr-x. 1 admin admin 2323 Jun 30 15:21 wp-comments-post.php
-rwxr-xr-x. 1 admin admin 3339 Jun 30 15:21 wp-config-sample.php
-rw-r--r--. 1 admin admin 301 Jun 30 15:24 wp-config.php
drwxr-xr-x. 4 admin admin 52 May 19 05:12 wp-content
-rwxr-xr-x. 1 admin admin 5617 Jun 30 15:21 wp-cron.php
drwxr-xr-x. 35 admin admin 16384 May 20 17:39 wp-includes
-rwxr-xr-x. 1 admin admin 2493 Jun 30 15:21 wp-links-opml.php
-rwxr-xr-x. 1 admin admin 3937 Jun 30 15:21 wp-load.php
-rwxr-xr-x. 1 admin admin 51850 Jun 30 15:21 wp-login.php
-rwxr-xr-x. 1 admin admin 8727 Jun 30 15:21 wp-mail.php
-rwxr-xr-x. 1 admin admin 32650 Jun 30 15:21 wp-settings.php
-rwxr-xr-x. 1 admin admin 34621 Jun 30 15:21 wp-signup.php
-rwxr-xr-x. 1 admin admin 5214 Jun 30 15:21 wp-trackback.php
-rwxr-xr-x. 1 admin admin 3205 Jun 30 15:21 xmlrpc.php
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Database Creation and Database User

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 10.5.29-MariaDB MariaDB Server

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)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]> SELECT User FROM mysql.user;
+-----+
| User |
+-----+
| mariadb.sys |
| mysql |
| root |
| wpuser |
+-----+
4 rows in set (0.001 sec)

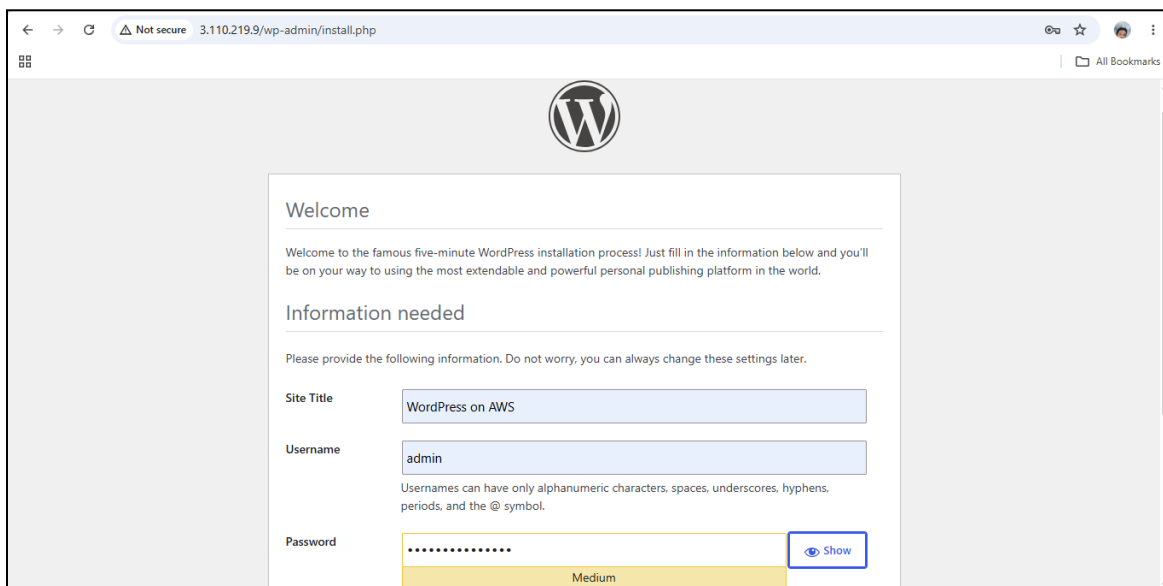
MariaDB [(none)]> █
```

(i) Verify Wordpress Deployment

Open the server IP address in a web browser to verify that the WordPress installation wizard is accessible.

URL: <http://<EC2-Public-IP>> or <http://3.110.219.9>

(ii) Complete Initial Wordpress Setup



The screenshot shows the WordPress installation wizard in a web browser. The browser's address bar displays "Not secure 3.110.219.9/wp-admin/install.php". The WordPress logo is centered at the top. Below it, a "Welcome" message reads: "Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world." The "Information needed" section follows, with the instruction: "Please provide the following information. Do not worry, you can always change these settings later." There are three input fields: "Site Title" with the value "WordPress on AWS", "Username" with the value "admin", and "Password" with masked characters. A "Show" button is next to the password field. Below the password field, a strength indicator shows "Medium".

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site
It is up to search engines to honor this request.

[Install WordPress](#)

→ Verify Successful Installation



Success!

WordPress has been installed. Thank you, and enjoy!

Username admin

Password *Your chosen password.*

[Log In](#)

→ Login to Wordpress Dashboard



Username or Email Address

Password

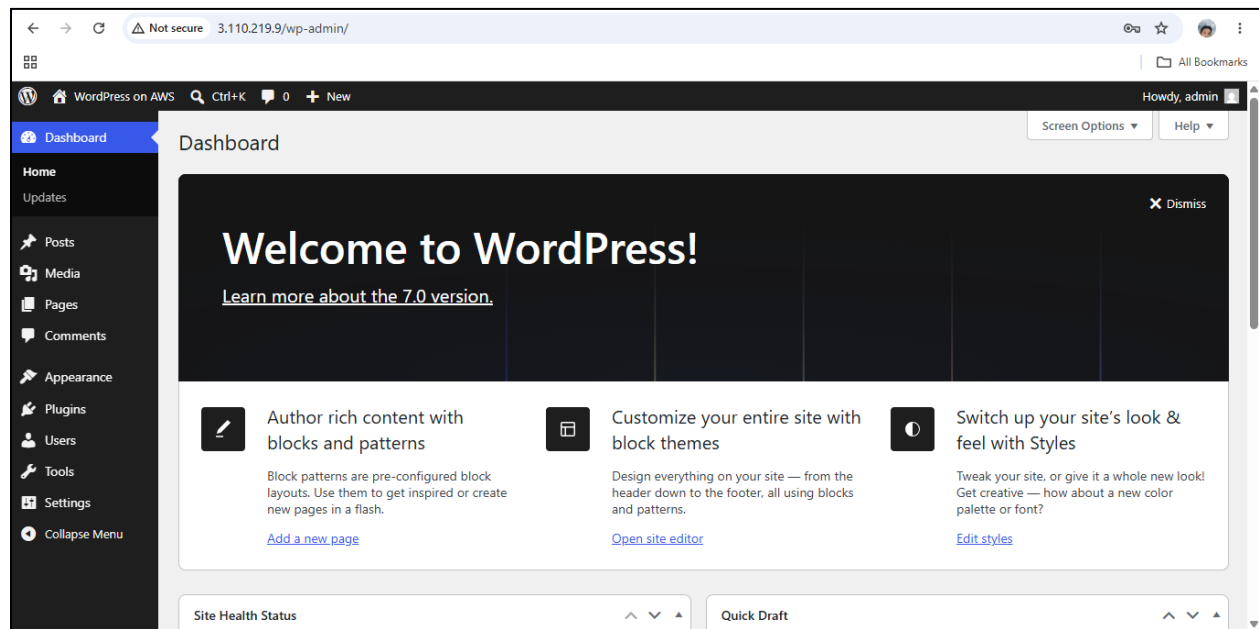


☐ Remember Me [Log In](#)

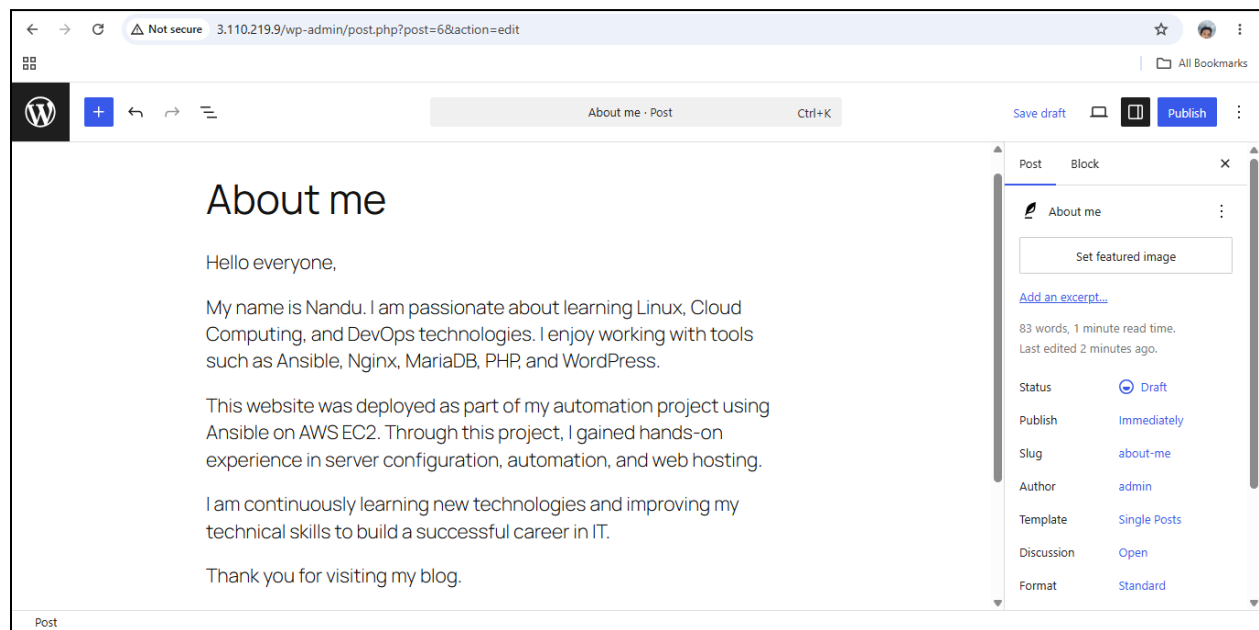
[Lost your password?](#)

[← Go to WordPress on AWS](#)

→ Access Wordpress Dashboard



→ Create Personal Blog Post



★ A personal blog post was created and published successfully through the Wordpress Dashboard.

Step 06: Phpmyadmin Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim phpmyadmin/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat phpmyadmin/tasks/main.yml
---
# tasks file for phpmyadmin

# phpMyAdmin Download
- name: Download phpMyAdmin
  get_url:
    url: https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
    dest: /tmp/phpmyadmin.tar.gz

# phpMyAdmin Extraction
- name: Extract phpMyAdmin
  unarchive:
    src: /tmp/phpmyadmin.tar.gz
    dest: /tmp
    remote_src: yes

# phpMyAdmin Deployment
- name: Copy phpMyAdmin Files
  copy:
    src: /tmp/phpMyAdmin-5.2.3-all-languages/
    dest: /home/admin/personalblog/public/phpmyadmin/
    remote_src: yes

# phpMyAdmin Nginx Configuration
- name: Create phpMyAdmin Nginx Config
  file:
    path: /etc/nginx/conf.d/phpmyadmin.conf
    state: touch
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Execute Phpmyadmin Role

```
TASK [phpmyadmin : Download phpMyAdmin] *****
changed: [localhost]

TASK [phpmyadmin : Extract phpMyAdmin] *****
changed: [localhost]

TASK [phpmyadmin : Copy phpMyAdmin Files] *****
changed: [localhost]

TASK [phpmyadmin : Create phpMyAdmin Nginx Config] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=22  changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

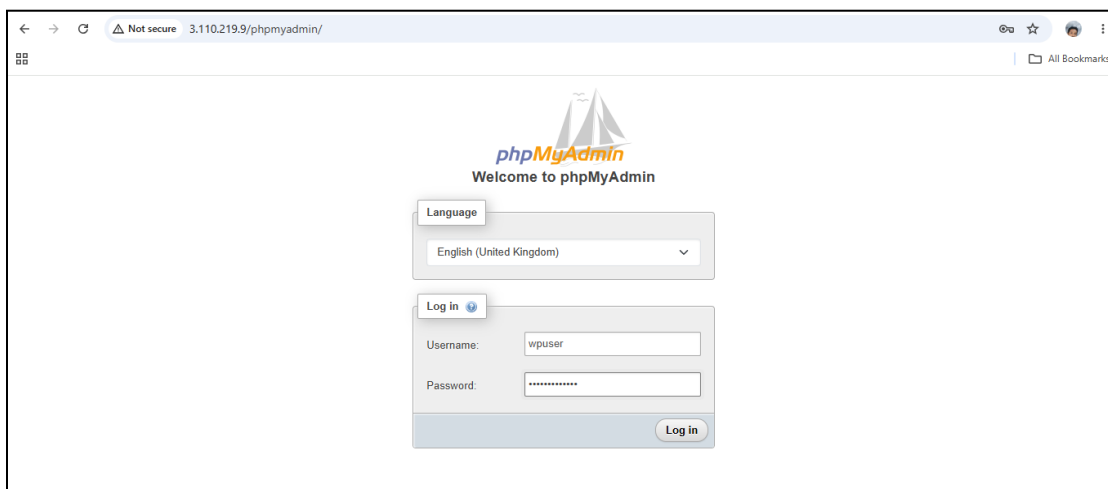
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

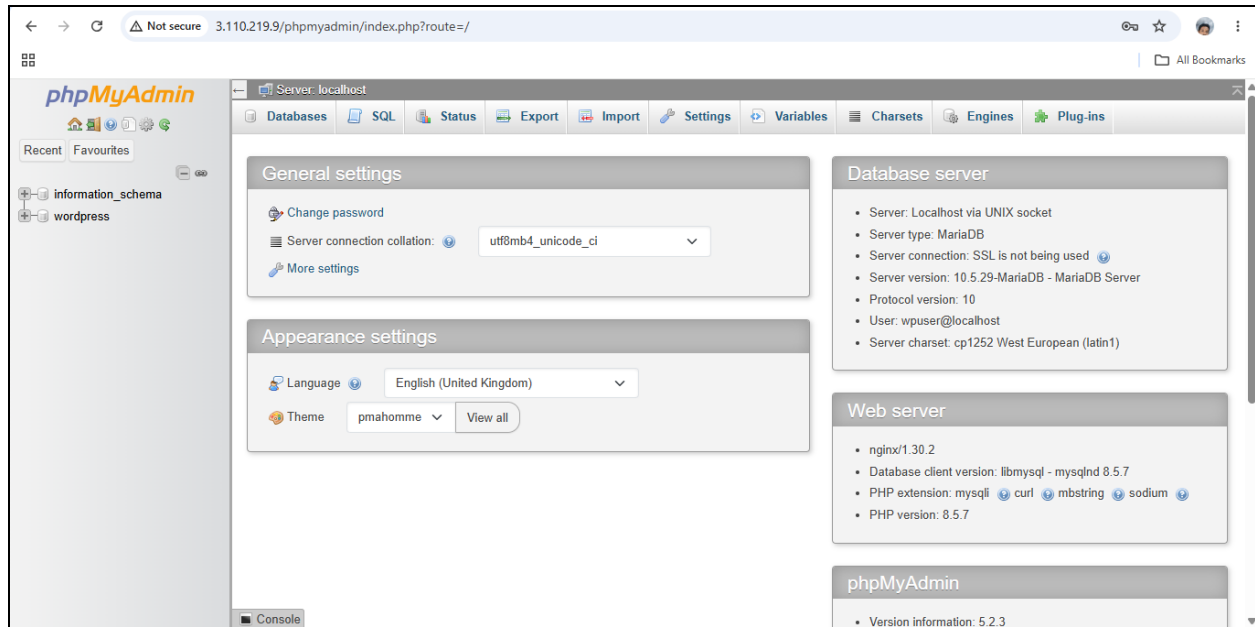
→ Verify Phpmyadmin Deployment

Access the **phpMyAdmin** URL and verify that the login page is displayed successfully.

URL: <http://65.1.111.213 /phpmyadmin>

→ Verify Phpmyadmin Dashboardn Access





Step 07: SFTP Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim sftp/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat sftp/tasks/main.yml
---
# tasks file for sftp
- name: Create admin user
  user:
    name: admin
    shell: /bin/bash
    create_home: yes
    state: present

- name: Set website ownership
  file:
    path: /home/admin/personalblog
    owner: admin
    group: admin
    recurse: yes
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Execute SFTP Role

```
TASK [sftp : Create admin user] *****
ok: [localhost]

TASK [sftp : Set website ownership] *****
ok: [localhost]

PLAY RECAP *****
localhost : ok=24  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify SFTP User and Permissions



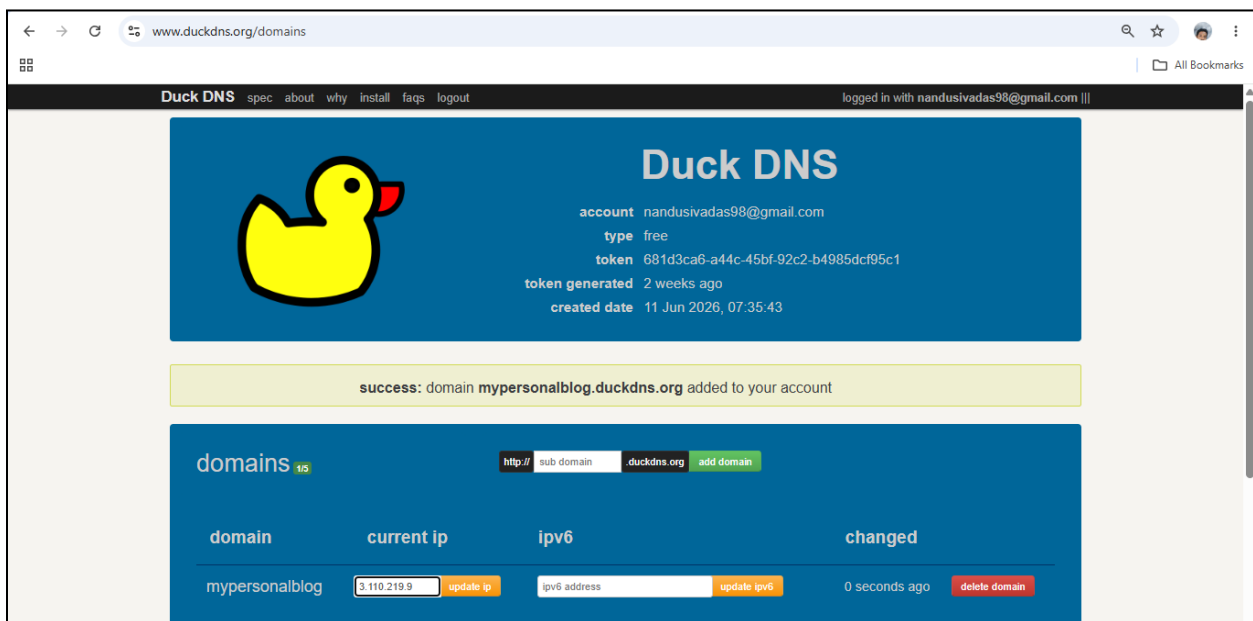
```
[ec2-user@ip-172-31-13-38 wordpress-project]$ id admin
uid=1001(admin) gid=1001(admin) groups=1001(admin)
[ec2-user@ip-172-31-13-38 wordpress-project]$ ls -ld /home/admin/personalblog
drwxr-xr-x. 3 admin admin 20 Jun 30 15:21 /home/admin/personalblog
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo passwd admin
Changing password for user admin.
New password:
BAD PASSWORD: The password contains the user name in some form
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl status sshd
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2026-06-30 14:44:05 UTC; 1h 43min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1873 (sshd)
    Tasks: 1 (limit: 1059)
  Memory: 4.3M
    CPU: 112ms
   CGroup: /system.slice/sshd.service
           └─1873 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jun 30 14:44:05 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Starting sshd.service - OpenSSH server daemon...
Jun 30 14:44:05 ip-172-31-13-38.ap-south-1.compute.internal sshd[1873]: Server listening on 0.0.0.0 port 22.
Jun 30 14:44:05 ip-172-31-13-38.ap-south-1.compute.internal sshd[1873]: Server listening on :: port 22.
Jun 30 14:44:05 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Started sshd.service - OpenSSH server daemon.
Jun 30 14:45:53 ip-172-31-13-38.ap-south-1.compute.internal sshd[2002]: Accepted publickey for ec2-user from 117.221.199.128 port 644
Jun 30 14:45:53 ip-172-31-13-38.ap-south-1.compute.internal sshd[2002]: pam_unix(sshd:session): session opened for user ec2-user(uid=
Jun 30 15:23:55 ip-172-31-13-38.ap-south-1.compute.internal sshd[30929]: error: kex_exchange_identification: Connection closed by rem
Jun 30 15:23:55 ip-172-31-13-38.ap-south-1.compute.internal sshd[30929]: Connection closed by 18.221.100.43 port 45896
Jun 30 16:02:21 ip-172-31-13-38.ap-south-1.compute.internal sshd[33194]: Invalid user from 8.148.235.102 port 44998
Jun 30 16:02:21 ip-172-31-13-38.ap-south-1.compute.internal sshd[33194]: Connection closed by invalid user 8.148.235.102 port 44998
lines 1-22/22 (END)
```

Domain Configuration

(i) Configure DNS Records

Configured the domain DNS settings by creating an a Record that points the domain name to the AWS EC2 public IP address.



The screenshot shows the Duck DNS website interface. At the top, there's a navigation bar with links like 'spec', 'about', 'why', 'install', 'faqs', and 'logout'. The user is logged in as 'nandusivadas98@gmail.com'. The main content area features a large blue box with a yellow duck logo and account details: account 'nandusivadas98@gmail.com', type 'free', token '681d3ca6-a44c-45bf-92c2-b4985dcf95c1', token generated '2 weeks ago', and created date '11 Jun 2026, 07:35:43'. Below this, a green box displays a success message: 'success: domain mypersonalblog.duckdns.org added to your account'. The bottom section, titled 'domains', shows a table with columns for 'domain', 'current ip', 'ipv6', and 'changed'. The table lists the domain 'mypersonalblog' with a current IP of '3.110.219.9' and an IPv6 address field. There are buttons for 'update ip', 'update ipv6', and 'delete domain'.

(ii) Configure Nginx Virtual Host

Configured the Nginx virtual host file and defined the domain name using the `server_name` directive

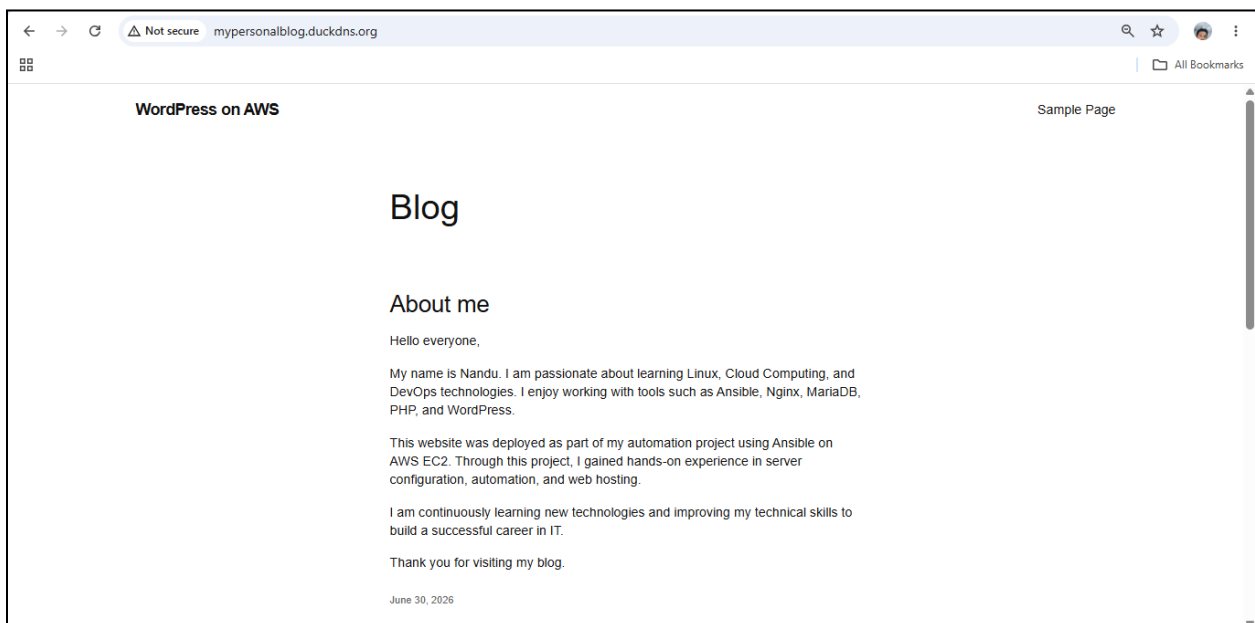
```
ec2-user@ip-172-31-13-38:~/wordpress-project
server {
    listen 80;
    server_name _mypersonalblog.duckdns.org;
}
```

(iii) Validate and Reload Nginx Configuration

Validated the Nginx configuration syntax and reloaded the service to apply the changes..

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo vim /etc/nginx/conf.d/wordpress.conf
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

(iv) Verify Domain Access



Step 08: SSL Configuration

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ vim ssl/tasks/main.yml
[ec2-user@ip-172-31-13-38 wordpress-project]$ cat ssl/tasks/main.yml
---
# tasks file for ssl
- name: Install Certbot
  dnf:
    name:
      - certbot
      - python3-certbot-nginx
    state: present

- name: Obtain SSL Certificate
  command: >
    certbot --nginx
    --non-interactive
    --agree-tos
    --email nandusivadas98@gmail.com
    -d mypersonalblog.duckdns.org

- name: Restart Nginx
  service:
    name: nginx
    state: restarted
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Execute SSL Role

```
TASK [ssl : Obtain SSL Certificate] *****
changed: [localhost]

TASK [ssl : Restart Nginx] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=27  changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Generate SSL Certificate Using Certbot

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo certbot install --cert-name mypersonalblog.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Deploying certificate
Successfully deployed certificate for mypersonalblog.duckdns.org to /etc/nginx/conf.d/wordpress.conf
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo certbot certificates
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Found the following certs:
Certificate Name: mypersonalblog.duckdns.org
Serial Number: 5090cb554aa6ab6a7e6f7b72320ea4501e8
Key Type: ECDSA
Domains: mypersonalblog.duckdns.org
Expiry Date: 2026-09-28 15:45:50+00:00 (VALID: 89 days)
Certificate Path: /etc/letsencrypt/live/mypersonalblog.duckdns.org/fullchain.pem
Private Key Path: /etc/letsencrypt/live/mypersonalblog.duckdns.org/privkey.pem
-----
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify SSL Certificate Files

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo ls -l /etc/letsencrypt/live/mypersonalblog.duckdns.org/
total 4
-rw-r--r--. 1 root root 692 Jun 30 16:44 README
lrwxrwxrwx. 1 root root 50 Jun 30 16:44 cert.pem -> ../../archive/mypersonalblog.duckdns.org/cert1.pem
lrwxrwxrwx. 1 root root 51 Jun 30 16:44 chain.pem -> ../../archive/mypersonalblog.duckdns.org/chain1.pem
lrwxrwxrwx. 1 root root 55 Jun 30 16:44 fullchain.pem -> ../../archive/mypersonalblog.duckdns.org/fullchain1.pem
lrwxrwxrwx. 1 root root 53 Jun 30 16:44 privkey.pem -> ../../archive/mypersonalblog.duckdns.org/privkey1.pem
[ec2-user@ip-172-31-13-38 wordpress-project]$
```

→ Verify Configure Nginx SSL Virtual Host

```
server
    listen 443 ssl;
    server_name mypersonalblog.duckdns.org;

    ssl_certificate /etc/letsencrypt/live/mypersonalblog.duckdns.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/mypersonalblog.duckdns.org/privkey.pem;

    root /home/admin/personalblog/public;
    index index.php index.html;

    location / {
        try_files $uri $uri/ /index.php?$args;
    }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_pass unix:/run/php-fpm/www.sock;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
```

Task 05 - Execute Main Playbook

❖ Run Complete Automation

ansible-playbook -i inventory site.yml

```
[ec2-user@ip-172-31-13-38 wordpress-project]$ ansible-playbook -i inventory site.yml

PLAY [web] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python3.9, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.15/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]

TASK [nginx : Install Nginx] *****
ok: [localhost]

TASK [nginx : Start and Enable Nginx] *****
ok: [localhost]

TASK [php : Install PHP] *****
ok: [localhost]

TASK [php : Start PHP-FPM] *****
ok: [localhost]

TASK [mariadb : Install MariaDB Server] *****
ok: [localhost]

TASK [mariadb : Install pip] *****
ok: [localhost]

TASK [mariadb : Install PyMySQL] *****
ok: [localhost]

TASK [mariadb : Start and Enable MariaDB] *****
ok: [localhost]

TASK [wordpress : Create Website User] *****
ok: [localhost]
```

```

TASK [wordpress : Create Website Directory] *****
changed: [localhost]

TASK [wordpress : Create WordPress Database] *****
ok: [localhost]

TASK [wordpress : Create WordPress Database User] *****
ok: [localhost]

TASK [wordpress : Download WordPress] *****
ok: [localhost]

TASK [wordpress : Extract WordPress] *****
ok: [localhost]

TASK [wordpress : Copy WordPress Files] *****
ok: [localhost]

TASK [wordpress : Configure wp-config.php] *****
changed: [localhost]

TASK [wordpress : Set Permissions] *****
ok: [localhost]

TASK [phpmyadmin : Download phpMyAdmin] *****
ok: [localhost]

TASK [phpmyadmin : Extract phpMyAdmin] *****
ok: [localhost]

TASK [phpmyadmin : Copy phpMyAdmin Files] *****
ok: [localhost]

TASK [phpmyadmin : Create phpMyAdmin Nginx Config] *****
changed: [localhost]

TASK [sftp : Create admin user] *****
ok: [localhost]

TASK [sftp : Set website ownership] *****
ok: [localhost]

TASK [ssl : Install Certbot] *****
ok: [localhost]

TASK [ssl : Obtain SSL Certificate] *****
changed: [localhost]

TASK [ssl : Restart Nginx] *****
changed: [localhost]

PLAY RECAP *****
localhost                : ok=27  changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[ec2-user@ip-172-31-13-38 wordpress-project]$

```

→ Verify Running Services

```

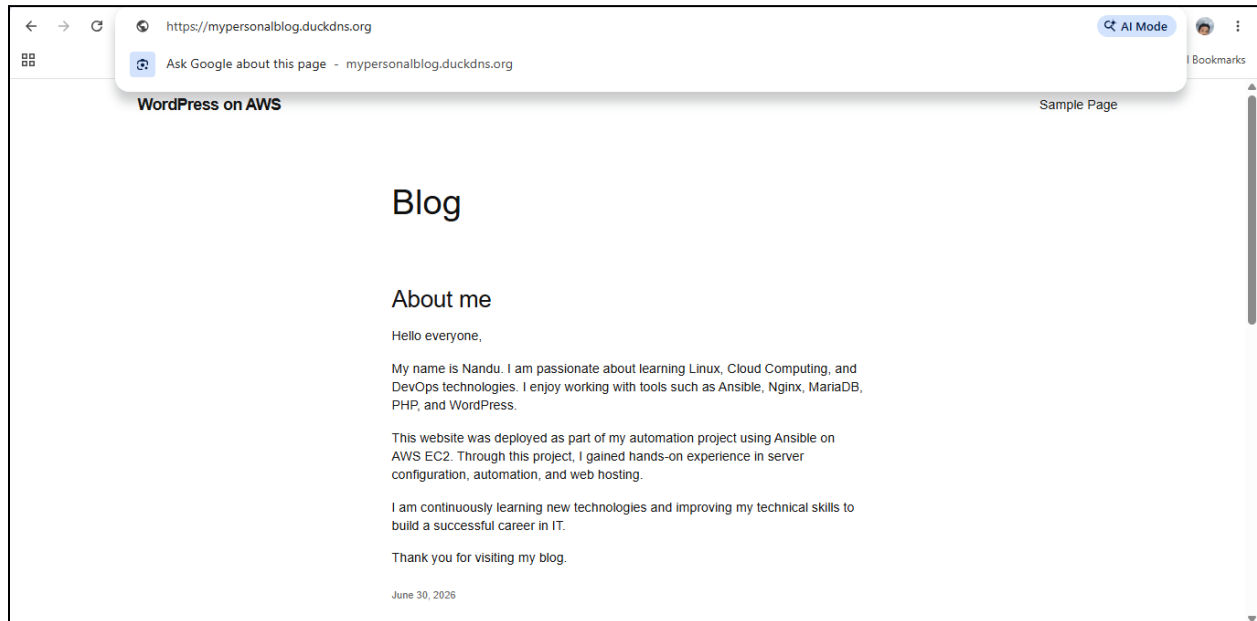
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl restart nginx
[ec2-user@ip-172-31-13-38 wordpress-project]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/nginx.service.d
           └─php-fpm.conf
   Active: active (running) since Tue 2026-06-30 16:56:25 UTC; 8s ago
     Process: 38410 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 38411 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 38412 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 38413 (nginx)
      Tasks: 3 (limit: 1059)
     Memory: 3.9M
        CPU: 64ms
    CGroup: /system.slice/nginx.service
            └─38413 "nginx: master process /usr/sbin/nginx"
              └─38414 "nginx: worker process"
                └─38415 "nginx: worker process"

Jun 30 16:56:25 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy ser
Jun 30 16:56:25 ip-172-31-13-38.ap-south-1.compute.internal nginx[38411]: nginx: the configuration file /etc/nginx/nginx.conf syntax >
Jun 30 16:56:25 ip-172-31-13-38.ap-south-1.compute.internal nginx[38411]: nginx: configuration file /etc/nginx/nginx.conf test is suc
Jun 30 16:56:25 ip-172-31-13-38.ap-south-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy serv

```

Executed the main Ansible playbook to deploy and configure the complete WordPress hosting environment, including Nginx, PHP, MariaDB, WordPress, phpMyAdmin, SFTP, Domain, and SSL configuration.

→ Verify HTTPS Website Access



SFTP Credentials

- Host : mypersonalblog.duckdns.org
- Username : admin
- Password : *****

PhpMyAdmin Credentials

- URL : http://mypersonalblog.duckdns.org/phpmyadmin
- Username : wpuser
- Password : *****

WordPress Credentials

- Admin URL : http://mypersonalblog.duckdns.org/wp-admin
- Username : admin
- Password : *****

Project Outcomes

The following deliverables were successfully completed:

- ✓ Ansible Project Initialization
- ✓ Inventory Configuration
- ✓ Nginx Installation and Configuration
- ✓ PHP Installation and Configuration
- ✓ MariaDB Installation and Database Setup
- ✓ Website User Creation and Permission Management
- ✓ WordPress Deployment and Configuration
- ✓ phpMyAdmin Deployment and Configuration
- ✓ SFTP User Configuration
- ✓ Domain Integration
- ✓ SSL Certificate Installation and HTTPS Enablement
- ✓ Main Playbook Execution
- ✓ End-to-End Verification and Validation

Final Outcome

The WordPress hosting environment was successfully deployed and automated using Ansible. All required services were configured correctly and verified through testing. The website was accessible through the configured domain name, WordPress administration panel was operational, phpMyAdmin was accessible for database management, SFTP access was successfully configured, and HTTPS encryption was enabled using SSL certificates.

Skills Demonstrated

- Linux System Administration
- AWS EC2
- Ansible Automation
- Infrastructure as Code (IaC)
- Nginx Web Server Configuration
- MariaDB Database Administration
- WordPress Deployment & Configuration
- SSL/TLS Implementation
- Secure File Transfer (SFTP)
- System Troubleshooting

Future Enhancements

- Docker-based deployment
- Kubernetes orchestration
- Jenkins CI/CD integration
- Terraform provisioning
- Automated backup solution
- Monitoring using Prometheus and Grafana

Conclusion

This project successfully demonstrated the automation of a secure WordPress hosting environment on AWS EC2 using Ansible Roles and Infrastructure as Code (IaC). The automated deployment reduced manual configuration, improved consistency, and simplified system management. All core components—including Nginx, PHP, MariaDB, WordPress, phpMyAdmin, SFTP, domain configuration, and SSL—were successfully deployed, configured, and validated. The project showcases practical Linux administration, cloud deployment, and automation skills applicable to real-world DevOps environments.

References

- AWS Documentation – <https://docs.aws.amazon.com/>
- Ansible Documentation – <https://docs.ansible.com/>
- Nginx Documentation – <https://nginx.org/en/docs/>
- PHP Documentation – <https://www.php.net/docs.php>
- MariaDB Documentation – <https://mariadb.com/kb/en/>
- WordPress Documentation – <https://developer.wordpress.org/>
- phpMyAdmin Documentation – <https://docs.phpmyadmin.net/>
- Certbot Documentation – <https://certbot.eff.org/docs/>
- Let's Encrypt Documentation – <https://letsencrypt.org/docs/>
- Duck DNS – <https://www.duckdns.org/>

GitHub Repository

Repository URL:

<https://github.com/nandusivadas/Automated-WordPress-Deployment-with-Ansible>

The complete source code, configuration files, documentation, and project resources are available in the GitHub repository.