

Main project - Using ansible

1. Install the necessary packages, which include MySQL 10.6, PHP version 8 or higher, and Nginx on your Linux server.
2. Create a user's home directory under the '/home' directory. You should set the root directory for the website to be '/home/username/website/public'."
3. Configure SFTP (Secure File Transfer Protocol) access for the user to ensure secure file management.
4. Configure phpmyadmin access for the user to ensure secure database management.
5. Install a free SSL certificate to enable secure HTTPS access to your website.
6. Once the website is hosted, proceed to the WordPress dashboard to make necessary configurations and customizations.
7. Write a personal blog post about yourself using the WordPress content management system.

What is Ansible

Ansible is an open-source automation tool used to manage and configure systems, deploy software, and orchestrate tasks across multiple machines. It uses simple, human-readable YAML (Yet Another Markup Language) files to define automation tasks, known as "playbooks." Ansible is agentless, meaning it doesn't require any special software or agent to be installed on the managed systems, making it easy to set up and use.

Key features of Ansible:

1. **Configuration Management:** Automates the configuration of systems, ensuring they are consistent and compliant with defined policies.
2. **Application Deployment:** Facilitates the deployment of applications to multiple servers or environments.
3. **Orchestration:** Manages complex workflows, such as the orchestration of tasks across different systems and services.
4. **Security and Compliance:** Helps enforce security policies by automating system configurations and updates.
5. **Idempotency:** Ensures that tasks can be safely run multiple times without causing unintended side effects.

1. Install nginx

```
[root@ip-192-168-1-227 ~]# yum install nginx -y
Last metadata expiration check: 0:06:12 ago on Sun Jan  5 14:04:49 2025.
Dependencies resolved.
=====
Package                        Architecture      Version           Repository        Size
=====
Installing:
nginx                          x86_64            1:1.26.2-1.amzn2023.0.1  amazonlinux      33 k
Installing dependencies:
generic-logos-httpd           noarch            18.0.0-12.amzn2023.0.3  amazonlinux      19 k
gperftools-libs               x86_64            2.9.1-1.amzn2023.0.3    amazonlinux      308 k
libunwind                     x86_64            1.4.0-5.amzn2023.0.2    amazonlinux      66 k
nginx-core                     x86_64            1:1.26.2-1.amzn2023.0.1  amazonlinux      670 k
nginxfilesystem               noarch            1:1.26.2-1.amzn2023.0.1  amazonlinux      9.9 k
nginx-mimetypes               noarch            2.1.49-3.amzn2023.0.3    amazonlinux      21 k
=====
```

2. Start and enable nginx

```
[root@ip-192-168-1-227 ~]# systemctl start nginx.service
[root@ip-192-168-1-227 ~]# systemctl enable nginx.service
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
```

3. Add a user and password

```
[root@ip-192-168-1-227 ~]# useradd vishnu
[root@ip-192-168-1-227 ~]# passwd vishnu
Changing password for user vishnu.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

4. Create a home directory

```
[root@ip-192-168-1-227 ~]# mkdir /home/vishnu/website
[root@ip-192-168-1-227 ~]# mkdir /home/vishnu/website/public
```

5. Create a nginx configuration file

```
[root@ip-192-168-1-227 ~]# vim /etc/nginx/conf.d/myweb.conf
```

```
server {
    listen      80;
    listen      [::]:80;
    server_name _;
    root        /home/Vishnu/website/public;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

6. Add permission to home directory

```
[root@ip-192-168-1-227 ~]# chmod -R 755 /home/vishnu
```

7. Restart nginx

```
[root@ip-192-168-1-227 ~]# systemctl restart nginx
```

8. Create a repo file for mariadb 10.6

```
[root@ip-192-168-1-227 ~]# vim /etc/yum.repos.d/MariaDB.repo
```

```
root@ip-192-168-1-227:~
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.6/rhel9-amd64
gpgkey = https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck = 1
enabled = 1
```

9. Install mysql and php

```
[root@ip-192-168-1-227 ~]# yum install php php-mysqld php-mysqli php-gd php-mbstring mariadb-server mariadb -y
MariaDB                               392 kB/s | 598 kB    00:01
Dependencies resolved.
=====
Package                                Architecture      Version           Repository        Size
=====
Installing:
MariaDB-client                         x86_64            10.6.20-1.el9     mariadb           8.5 M
MariaDB-server                         x86_64            10.6.20-1.el9     mariadb           18 M
php8.3                                x86_64            8.3.10-1.amzn2023.0.1 amazonlinux       10 k
php8.3-gd                             x86_64            8.3.10-1.amzn2023.0.1 amazonlinux       43 k
php8.3-mbstring                       x86_64            8.3.10-1.amzn2023.0.1 amazonlinux       528 k
php8.3-mysqld                         x86_64            8.3.10-1.amzn2023.0.1 amazonlinux       147 k
=====
```

10. Start and enable mariadb

```
[root@ip-192-168-1-227 ~]# systemctl start mariadb
[root@ip-192-168-1-227 ~]# systemctl enable mariadb
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
```

11. Mariadb-secure-installation

```
[root@ip-192-168-1-227 ~]# mariadb-secure-installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n]
Enabled successfully!
Reloading privilege tables..
... Success!
```

12. Create database

```
[root@ip-192-168-1-227 ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.6.20-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)]> CREATE DATABASE vishnu_db;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'vishnu'@'localhost' IDENTIFIED BY 'redhat';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON vishnu_db.* TO 'vishnu'@'localhost';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES
-> FLUSH PRIVILEGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to the right syntax to use near 'FLUSH PRIVILEGES' at line 2
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> EXIT;
Bye
```

13. Change directory

```
[root@ip-192-168-1-227 ~]# cd /home/vishnu/website/public/
[root@ip-192-168-1-227 public]#
```

14. Install wordpress

```
[root@ip-192-168-1-227 public]# wget https://wordpress.org/latest.zip
--2025-01-05 14:35:26-- https://wordpress.org/latest.zip
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: 28585184 (27M) [application/zip]
Saving to: 'latest.zip'

latest.zip          100%[=====] 27.26M  6.44MB/s   in 5.7s
2025-01-05 14:35:33 (4.77 MB/s) - 'latest.zip' saved [28585184/28585184]
```

15. Unzip the file

```
[root@ip-192-168-1-227 public]# unzip latest.zip
Archive: latest.zip
  creating: wordpress/
  inflating: wordpress/xmlrpc.php
  inflating: wordpress/wp-blog-header.php
  inflating: wordpress/readme.html
  inflating: wordpress/wp-signup.php
  inflating: wordpress/index.php
  inflating: wordpress/wp-cron.php
  inflating: wordpress/wp-config-sample.php
  inflating: wordpress/wp-login.php
  inflating: wordpress/wp-settings.php
  inflating: wordpress/license.txt
  creating: wordpress/wp-content/
  creating: wordpress/wp-content/themes/
```

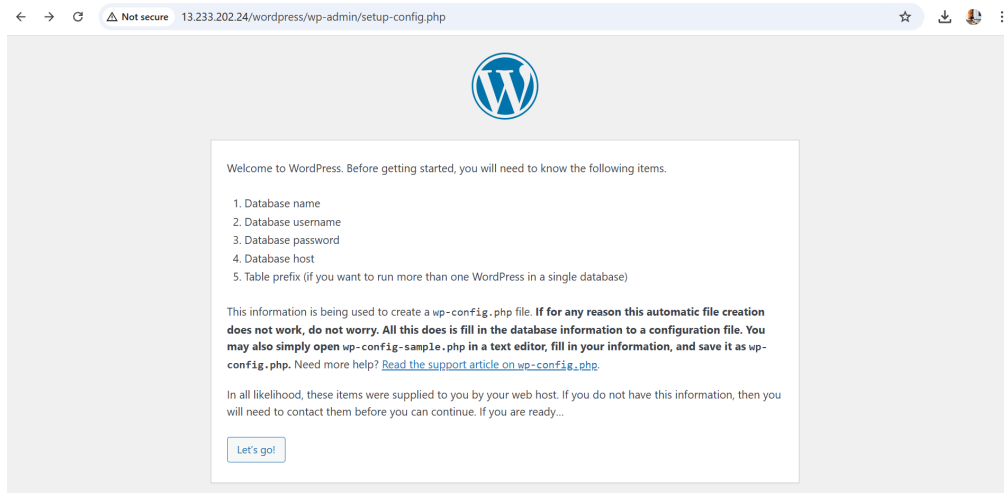
16. Remove the file

```
[root@ip-192-168-1-227 public]# rm -f latest.zip
```

17. Add permission to wp-content

```
[root@ip-192-168-1-227 public]# chmod -R 755 wordpress/wp-content/
```

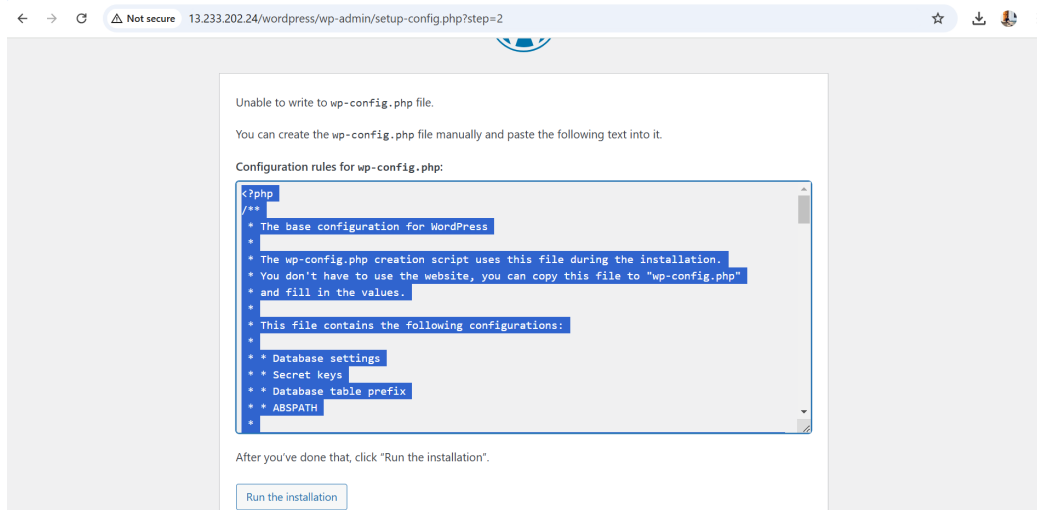
18. Open browser and search 'IP address/wordpress'



19. Fill the database details

A screenshot of the WordPress database configuration screen. The browser's address bar shows the URL '13.233.202.24/wordpress/wp-admin/setup-config.php?step=1'. The page contains a form with the following fields: 'Database Name' with the value 'vishnu_db', 'Username' with the value 'vishnu', 'Password' with the value 'redhat' and a 'Hide' button, 'Database Host' with the value 'localhost', and 'Table Prefix' with the value 'wp_'. Each field has a descriptive text below it. At the bottom of the form is a blue button labeled 'Submit'.

20. Copy the configuration file details



21. Create wp-config.php file and paste the configuration details

```
[root@ip-192-168-1-227 public]# cd wordpress
[root@ip-192-168-1-227 wordpress]# vim wp-config.php
```

```
root@ip-192-168-1-227/home x + v
* @package WordPress
*/

// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'vishnu_db' );

/** Database username */
define( 'DB_USER', 'vishnu' );

/** Database password */
define( 'DB_PASSWORD', 'redhat' );

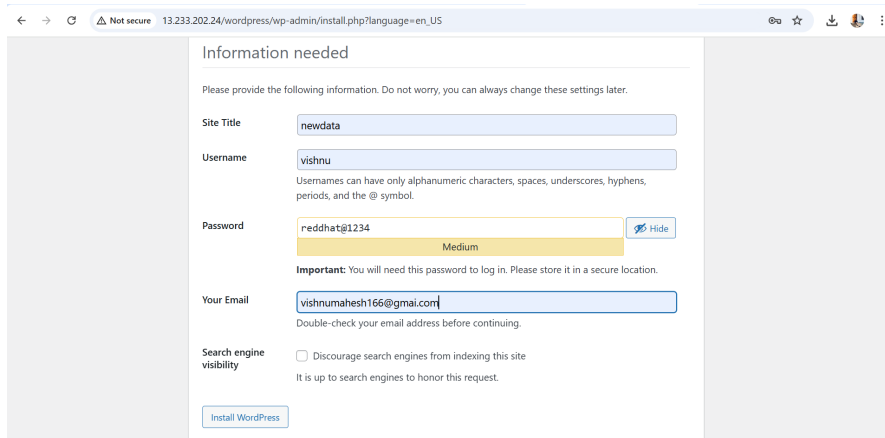
/** Database hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * You can change these at any point in time to invalidate all existing cookies.
```

22. Create user and site

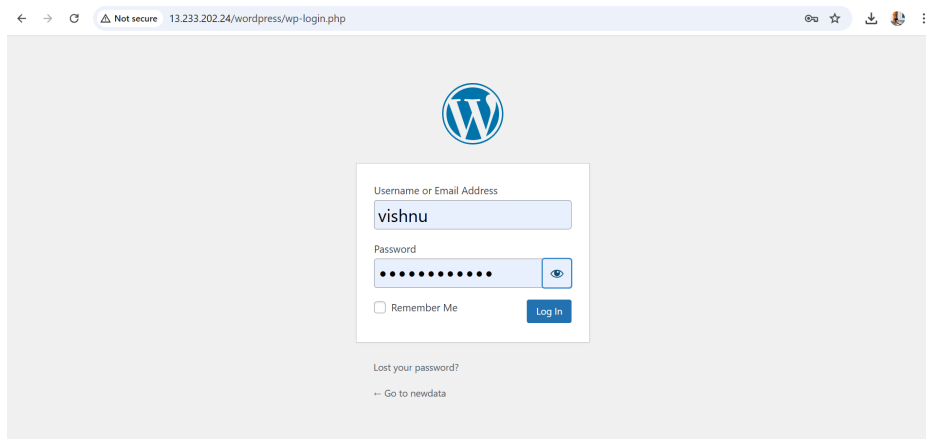


The screenshot shows the 'Information needed' step of the WordPress installation process. The browser address bar indicates the URL: 13.233.202.24/wordpress/wp-admin/install.php?language=en_US. The form contains the following fields and options:

- Site Title:** newdata
- Username:** vishnu. A note below states: 'Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.'
- Password:** reddhat@1234. A strength indicator shows 'Medium'. A 'Hide' button is present.
- Your Email:** vishnumahesh166@gmail.com. A note below states: 'Double-check your email address before continuing.'
- Search engine visibility:** An unchecked checkbox labeled 'Discourage search engines from indexing this site'. A note below states: 'It is up to search engines to honor this request.'

An 'Install WordPress' button is located at the bottom left of the form.

23. Log in to wordpress user

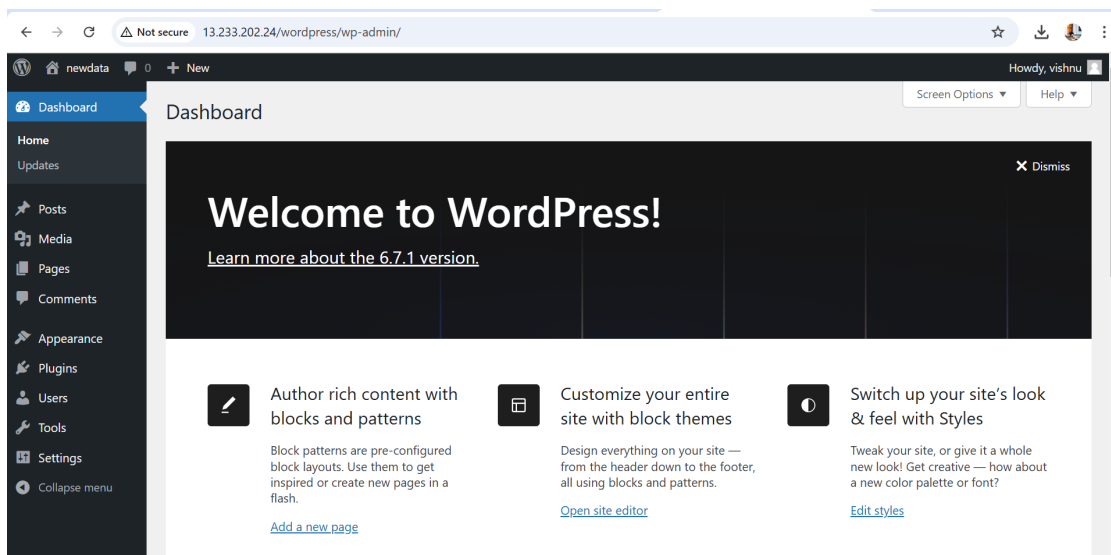


The screenshot shows the WordPress login page. The browser address bar indicates the URL: 13.233.202.24/wordpress/wp-login.php. The page features the WordPress logo at the top center. Below it is a login form with the following fields and options:

- Username or Email Address:** vishnu
- Password:** A masked password field with a visibility toggle icon.
- Remember Me:** An unchecked checkbox.
- Log In:** A blue button.

Below the login form, there is a link for 'Lost your password?' and a link to 'Go to newdata'.

24. Output



The screenshot shows the WordPress admin dashboard. The browser address bar indicates the URL: 13.233.202.24/wordpress/wp-admin/. The dashboard includes a left sidebar with navigation links: Dashboard, Home, Updates, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, Settings, and Collapse menu. The main content area displays a 'Welcome to WordPress!' message with a link to 'Learn more about the 6.7.1 version.' Below this, there are three featured sections:

- Author rich content with blocks and patterns:** A section describing block patterns and providing a link to 'Add a new page'.
- Customize your entire site with block themes:** A section describing block themes and providing a link to 'Open site editor'.
- Switch up your site's look & feel with Styles:** A section describing the Styles interface and providing a link to 'Edit styles'.

The dashboard also includes a top bar with 'Screen Options' and 'Help' buttons, and a user profile section showing 'Howdy, vishnu'.

25. Change directory

```
[root@ip-192-168-1-227 wordpress]# cd /home/vishnu/website/public
```

26. Install phpMyAdmin

```
[root@ip-192-168-1-227 public]# wget https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip
--2025-01-05 15:09:13-- https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip
Resolving files.phpmyadmin.net (files.phpmyadmin.net)... 89.187.163.111, 89.187.162.13, 79.127.235.8, ...
Connecting to files.phpmyadmin.net (files.phpmyadmin.net)|89.187.163.111|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15096155 (14M) [application/zip]
Saving to: 'phpMyAdmin-5.2.1-all-languages.zip'

phpMyAdmin-5.2.1-all-language 100%[=====] 14.40M 14.8MB/s in 1.0s

2025-01-05 15:09:14 (14.8 MB/s) - 'phpMyAdmin-5.2.1-all-languages.zip' saved [15096155/15096155]
```

27. Unzip the file

```
[root@ip-192-168-1-227 public]# unzip phpMyAdmin-5.2.1-all-languages.zip
Archive:  phpMyAdmin-5.2.1-all-languages.zip
  creating: phpMyAdmin-5.2.1-all-languages/
  extracting: phpMyAdmin-5.2.1-all-languages/.rtlcssrc.json
  inflating: phpMyAdmin-5.2.1-all-languages/CONTRIBUTING.md
  inflating: phpMyAdmin-5.2.1-all-languages/ChangeLog
  inflating: phpMyAdmin-5.2.1-all-languages/LICENSE
  inflating: phpMyAdmin-5.2.1-all-languages/README
  extracting: phpMyAdmin-5.2.1-all-languages/RELEASE-DATE-5.2.1
  extracting: phpMyAdmin-5.2.1-all-languages/babel.config.json
  inflating: phpMyAdmin-5.2.1-all-languages/composer.json
  inflating: phpMyAdmin-5.2.1-all-languages/composer.lock
  inflating: phpMyAdmin-5.2.1-all-languages/config.sample.inc.php
```

28. Copy the file to phpMyAdmin

```
[root@ip-192-168-1-227 public]# cp -r phpMyAdmin-5.2.1-all-languages phpMyAdmin
```

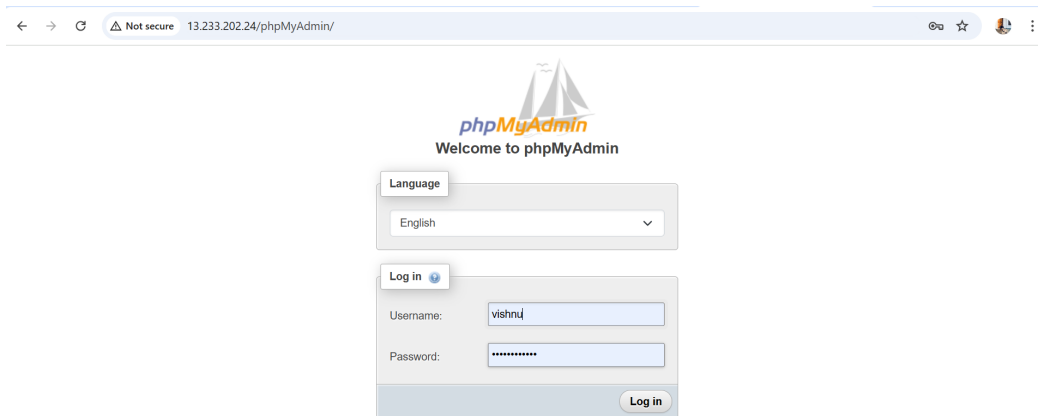
29. Remove the files

```
[root@ip-192-168-1-227 public]# rm -rf phpMyAdmin-5.2.1-all-languages
```

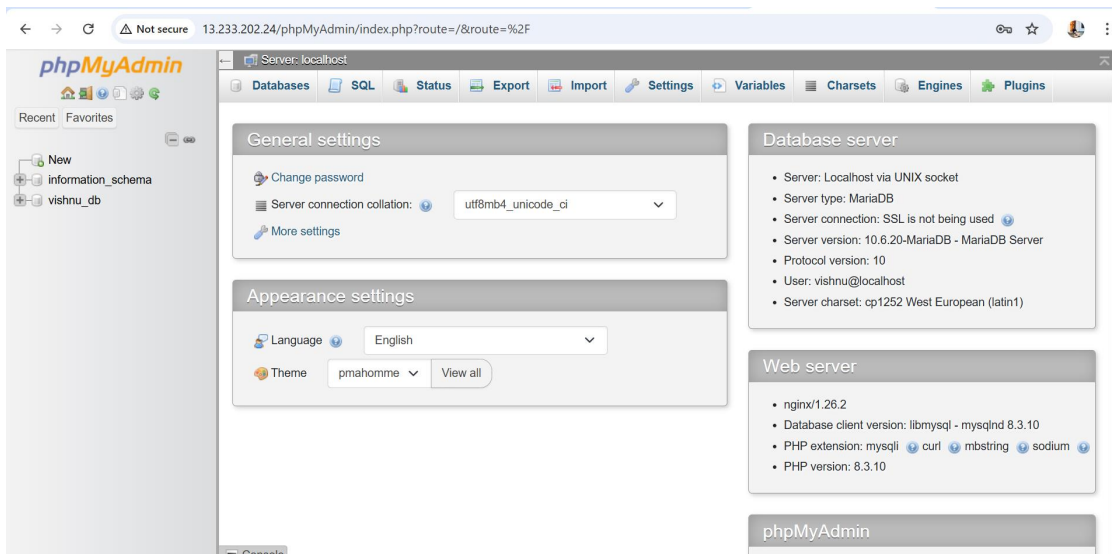
```
[root@ip-192-168-1-227 public]# rm -rf phpMyAdmin-5.2.1-all-languages.zip
```


30. Open browser and search 'IP address/phpMyAdmin'

Enter the user name and password



31. Output



32. Edit to add ssh configuration file

```
[root@ip-192-168-1-227 wordpress]# vim /etc/ssh/sshd_config
```

```
Match User vishnu
    ForceCommand internal-sftp
    PasswordAuthentication yes
"/etc/ssh/sshd_config" 137L, 3935B
```

33. Change ownership

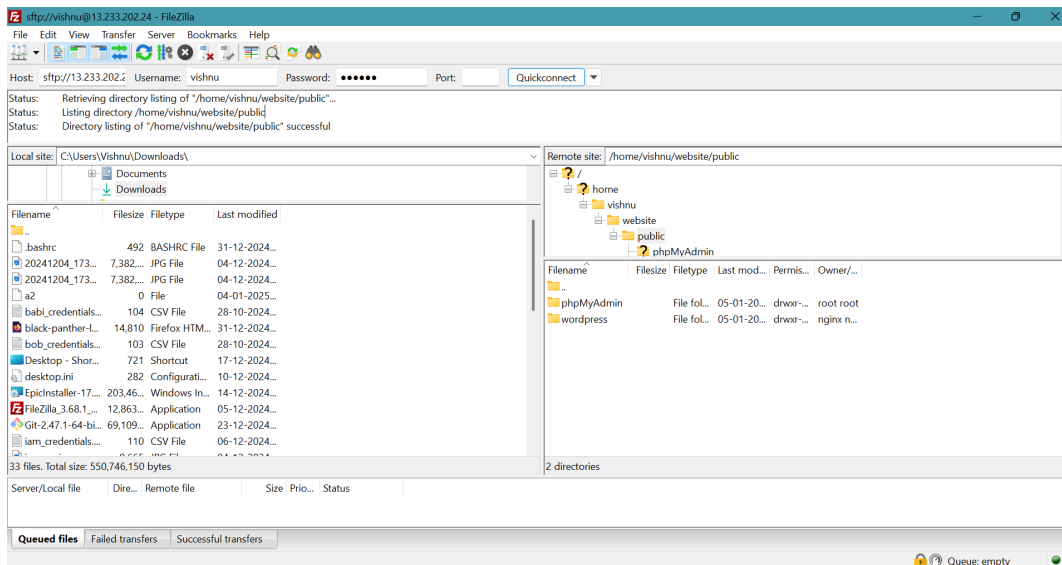
```
[root@ip-192-168-1-227 wordpress]# chown root:root /home/vishnu
```

```
[root@ip-192-168-1-227 wordpress]# chown vishnu:vishnu /home/vishnu/website/public
```

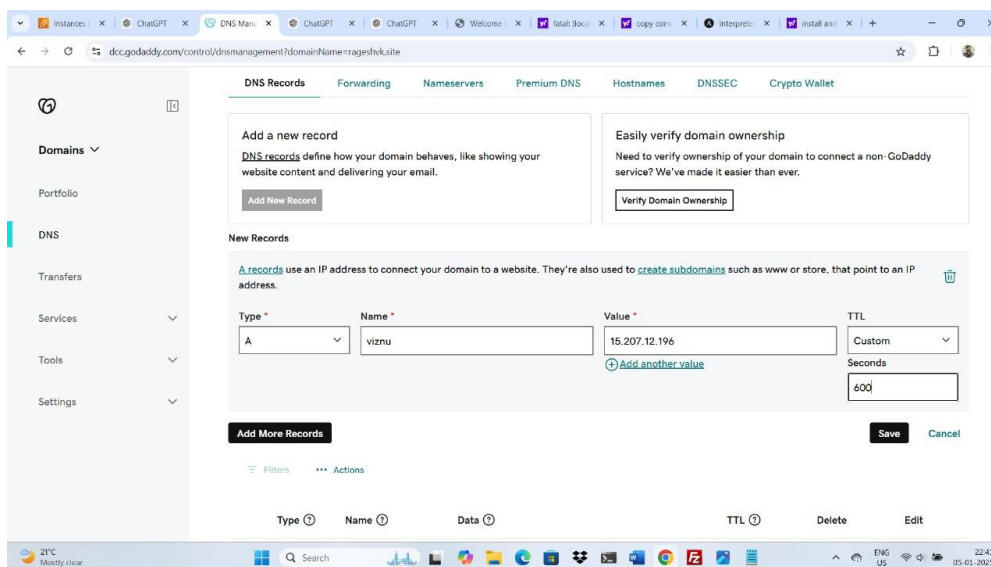
34. Restart sshd

```
[root@ip-192-168-1-227 wordpress]# systemctl restart sshd
```

35. Open filezilla



36. GoDaddy assigns domain name to IP address

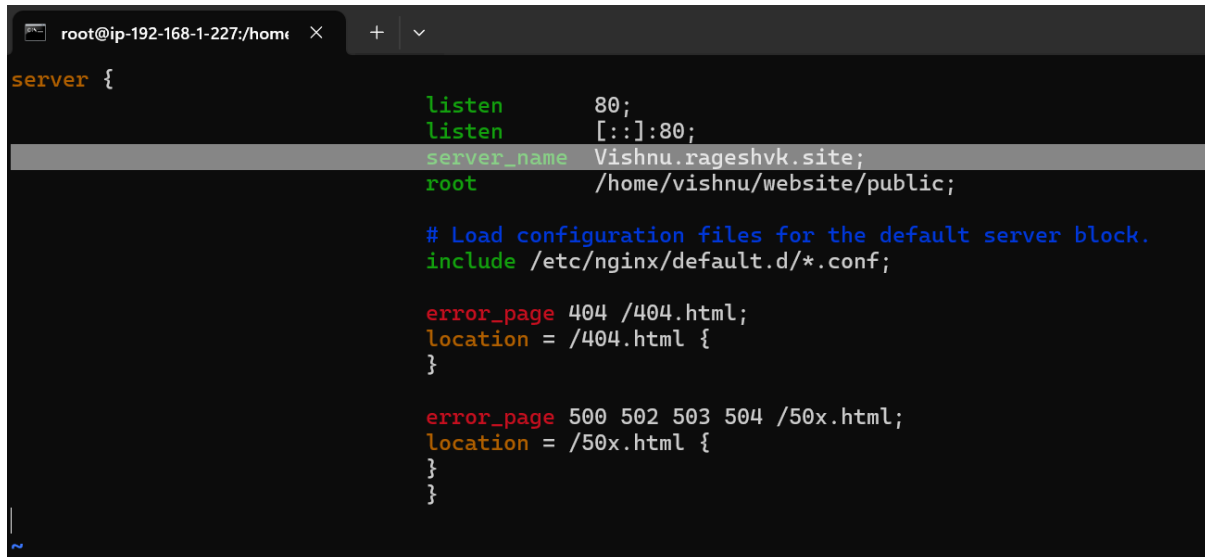


37. Install certbot

```
[root@ip-192-168-1-227 wordpress]# yum install certbot python3-certbot-nginx -y
Last metadata expiration check: 1:04:39 ago on Sun Jan  5 14:27:33 2025.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
certbot                                noarch            2.6.0-4.amzn2023.0.1  amazonlinux         49 k
python3-certbot-nginx                 noarch            2.6.0-4.amzn2023.0.1  amazonlinux        158 k
Installing dependencies:
fontawesome-fonts                     noarch            1:4.7.0-11.amzn2023.0.2  amazonlinux        205 k
python3-acme                          noarch            2.6.0-4.amzn2023.0.1  amazonlinux        161 k
python3-certbot                       noarch            2.6.0-4.amzn2023.0.1  amazonlinux        677 k
python3-configargparse                noarch            1.7-1.amzn2023         amazonlinux         45 k
python3-josepy                        noarch            1.13.0-6.amzn2023       amazonlinux         61 k
python3-parsedatetime                 noarch            2.6-10.amzn2023        amazonlinux         80 k
python3-pyOpenSSL                     noarch            21.0.0-1.amzn2023.0.2  amazonlinux         92 k
python3-pyparsing                     noarch            2.4.7-6.amzn2023.0.2  amazonlinux        152 k
python3-pyrfc3339                     noarch            1.1-16.amzn2023        amazonlinux         19 k
=====
```

38. Edit to add server name in /etc/nginx/conf.d/myweb.conf

```
[root@ip-192-168-1-227 wordpress]# vim /etc/nginx/conf.d/myweb.conf
```



```
server {
    listen      80;
    listen      [::]:80;
    server_name Vishnu.rageshvk.site;
    root        /home/vishnu/website/public;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

39. Test the Nginx configuration syntax

```
[root@ip-192-168-1-227 wordpress]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

40. Restart nginx

```
[root@ip-192-168-1-227 wordpress]# systemctl restart nginx
[root@ip-192-168-1-227 wordpress]#
```

41. Create SSL certificate

```
[root@ip-192-168-1-136 public]# certbot --nginx -d viznu.rageshvk.site
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
(Enter 'c' to cancel): vishnumahesh166@gmail.com

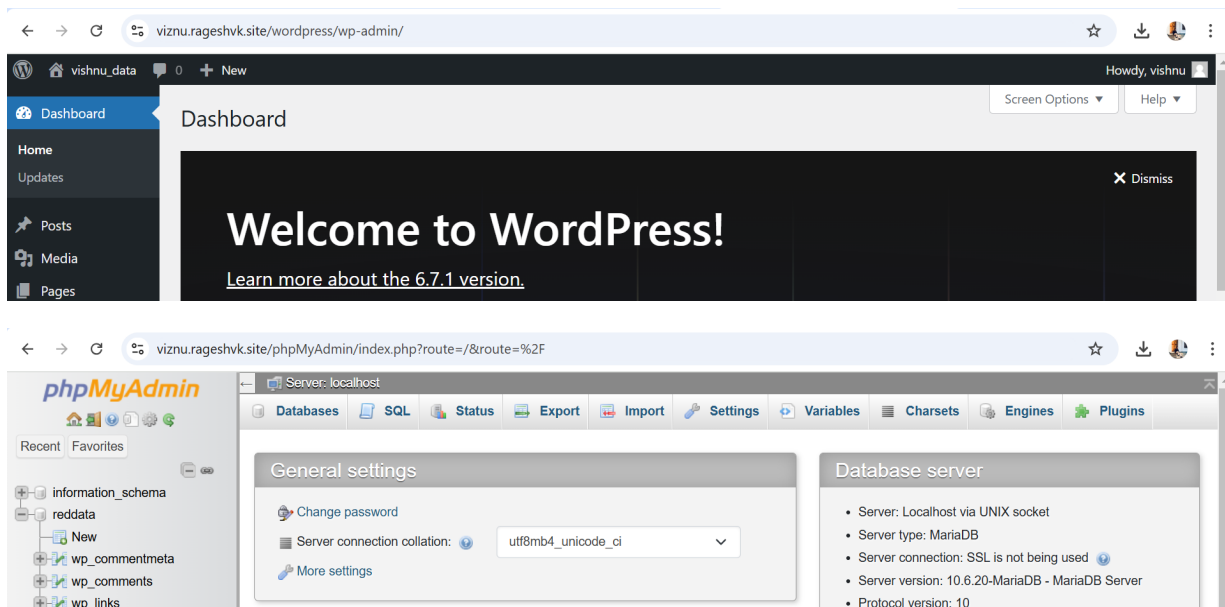
-----
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.4-April-3-2024.pdf. You must agree in
order to register with the ACME server. Do you agree?
-----
(Y)es/(N)o: y

-----
Would you be willing, once your first certificate is successfully issued, to
share your email address with the Electronic Frontier Foundation, a founding
partner of the Let's Encrypt project and the non-profit organization that
develops Certbot? We'd like to send you email about our work encrypting the web,
EFF news, campaigns, and ways to support digital freedom.
-----
(Y)es/(N)o: y
Account registered.
Requesting a certificate for viznu.rageshvk.site

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/viznu.rageshvk.site/fullchain.pem
Key is saved at: /etc/letsencrypt/live/viznu.rageshvk.site/privkey.pem
This certificate expires on 2025-04-05.
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 viznu.rageshvk.site to /etc/nginx/conf.d/mydata.conf
Congratulations! You have successfully enabled HTTPS on https://viznu.rageshvk.site
```

42. Search domain name



Credentials:

- SFTP Credentials:

- **Host:** viznu.rageshvk.site

- **Username:** [vishnu](#)

- **Password:** [redhat](#)

- phpMyAdmin Credentials:

- **URL:** viznu.rageshvk.site/phpMyAdmin

- **Username:** [viz](#)

- **Password:** [1234](#)

- WordPress Credentials:

- **WordPress Admin URL:** viznu.rageshvk.site/wordpress/wp-admin

- **Username:** [vishnu](#)

- **Password:** [redhat@vishnu](#)

- **personal Blog:** viznu.rageshvk.site/wordpress

USING ANSIBLE

1. Install ANSIBLE

```
[root@ip-192-168-1-87 ~]# yum install ansible -y
Last metadata expiration check: 0:08:16 ago on Mon Jan 6 17:31:34 2025.
Dependencies resolved.
=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
ansible                                noarch            8.3.0-1.amzn2023.0.1  amazonlinux         32 M
Installing dependencies:
ansible-core                          x86_64            2.15.3-1.amzn2023.0.4  amazonlinux         2.5 M
git-core                              x86_64            2.40.1-1.amzn2023.0.3  amazonlinux         4.3 M
sshpas                                x86_64            1.09-6.amzn2023.0.1    amazonlinux         28 k
=====
Transaction Summary
=====
Install 4 Packages
```

2. Create Inventory file

```
[root@ip-192-168-1-87 ~]# vim inventory
```

Add instance public IP Address

```
root@ip-192-168-1-190:~
15.206.93.101
```

3. Create template for wp-config.php file and its steps

```
[root@ip-192-168-1-87 ~]# vim wp-config.php.j2
```

```
root@ip-192-168-1-190:~ root@ip-192-168-1-87:~
<?php
// ** MySQL settings ** //
define('DB_NAME', 'vishnu_DB');
define('DB_USER', 'vishnu');
define('DB_PASSWORD', 'redhat');
define('DB_HOST', 'localhost');

// ** Authentication unique keys and salts. ** //
define('AUTH_KEY', 'your-auth-key');
define('SECURE_AUTH_KEY', 'your-secure-auth-key');
define('LOGGED_IN_KEY', 'your-logged-in-key');
define('NONCE_KEY', 'your-nonce-key');
define('AUTH_SALT', 'your-auth-salt');
define('SECURE_AUTH_SALT', 'your-secure-auth-salt');
define('LOGGED_IN_SALT', 'your-logged-in-salt');
define('NONCE_SALT', 'your-nonce-salt');

// ** WordPress Database Table prefix. ** //
$table_prefix = 'wp_';

// For developers: WordPress debugging mode.
define('WP_DEBUG', false);

/* That's all, stop editing! Happy publishing. */
define('ABSPATH', dirname(__FILE__) . '/');
require_once(ABSPATH . 'wp-settings.php');
```

```
<? php // ** MySQL settings ** //  
  
define( 'DB_NAME', 'vishnu_DB' );  
  
define( 'DB_USER', 'vishnu' );  
  
define( 'DB_PASSWORD', 'redhat' );  
  
define( 'DB_HOST', 'localhost' );  
  
// ** Authentication unique keys and salts. ** //  
  
define ( 'AUTH_KEY' , 'your-auth-key' );  
  
define ( ' SECURE_AUTH_KEY' 'your-secure-auth-key' );  
  
define( 'LOGGED_IN_KEY' , 'your-logged-in-key' );  
  
define ( 'NONCE_KEY' , 'your-nonce-key' );  
  
define ( ' AUTH_SALT', 'your-auth-salt' );  
  
define( 'SECURE_AUTH_SALT', 'your-secure-auth-salt' );  
  
define( 'LOGGED_IN_SALT' , 'your-logged-in-salt' );  
  
define( ' NONCE_SALT', 'your-nonce-salt' );  
  
// ** WordPress Database Table prefix. ** //  
  
$table_prefix = 'wp_';  
  
// For developers: WordPress debugging mode.  
  
define( 'WP_DEBUG', false );  
  
/* That's all, stop editing! Happy publishing. */  
  
define( 'ABSPATH', dirname( __ FILE__ ) . '/' );  
  
require_once (ABSPATH . 'wp-settings . php' );
```

4. Create Ansible play-book

Vim ansible.yml

```
---
- name: Set up MariaDB, PHP, Nginx, and configure website
  hosts: localhost
  become: yes
  tasks:
    - name: Ensure /etc/yum.repos.d directory exists
      file:
        path: /etc/yum.repos.d
        state: directory

    - name: Create MariaDB repository file
      copy:
        dest: /etc/yum.repos.d/MariaDB.repo
        content: |
          [mariadb]
          name = MariaDB
          baseurl = http://yum.mariadb.org/10.6/rhel9-amd64
          gpgkey = https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
          gpgcheck = 1

    - name: Ensure python3-pip is installed
      yum:
        name: python3-pip
        state: present

    - name: Install passlib library
      pip:
        name: passlib
        state: present

    - name: Install PHP, MariaDB server, and required PHP extensions
      yum:
        name:
          - php
          - php-mysqlnd
          - php-mysqli
          - php-gd
          - php-mbstring
          - mariadb-server
          - mariadb
        state: present

    - name: Install Nginx
      yum:
        name: nginx
        state: present

    - name: Start and enable MariaDB service
```



```

- name: Start and enable MariaDB service
  systemd:
    name: mariadb
    enabled: yes
    state: started

- name: Start and enable Nginx service
  systemd:
    name: nginx
    enabled: yes
    state: started

- name: Create a user named vishnu
  user:
    name: vishnu
    password: "{{ 'redhat' | password_hash('sha512') }}"
    create_home: yes

- name: Create website directory for vishnu
  file:
    path: /home/vishnu/website/public
    state: directory
    owner: vishnu
    group: vishnu
    mode: '0755'

- name: Configure Nginx server block for vishnu
  copy:
    dest: /etc/nginx/conf.d/mydata.conf
    content: |
      server {
        listen 80;
        listen [::]:80;
        server_name _; # Default server name

        root /home/vishnu/website/public;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
      }

```

```

    }

- name: Test Nginx configuration
  command: nginx -t
  register: nginx_test_result
  failed_when: "'syntax is okay' not in nginx_test_result.stderr and 'test is successful' not in nginx_test_result.stderr"

- name: Reload Nginx to apply changes
  systemd:
    name: nginx
    state: reloaded

- name: Set permissions for /home/vishnu
  command: chmod -R 755 /home/vishnu/
  become: yes

- name: Change ownership for /home/vishnu/website
  command: chown -R nginx:nginx /home/vishnu/website/
  become: yes

- name: Navigate to the website public directory
  shell: cd /home/vishnu/website/public
  args:
    chdir: /home/vishnu/website/public
  become: yes

- name: Download WordPress latest.zip
  get_url:
    url: https://wordpress.org/latest.zip
    dest: /home/vishnu/website/public/latest.zip

- name: Unzip the WordPress archive
  unarchive:
    src: /home/vishnu/website/public/latest.zip
    dest: /home/vishnu/website/public/
    remote_src: yes

- name: Remove the downloaded zip file
  file:
    path: /home/vishnu/website/public/latest.zip
    state: absent

- name: Set permissions for wp-content directory
  command: chmod -R 755 /home/vishnu/website/public/wordpress/wp-content
  become: yes

- name: Change ownership of wp-content directory
  command: chown -R nginx:nginx /home/vishnu/website/public/wordpress/wp-content

```

```

- name: Change ownership of wp-content directory
  command: chown -R nginx:nginx /home/vishnu/website/public/wordpress/wp-content
  become: yes

# Secure MariaDB installation using SQL commands, only if root password is not set
- name: Check if MariaDB root password is set
  command: mysqladmin -u root status
  register: mariadb_root_check
  failed_when: false
  changed_when: false

- name: Secure MariaDB installation
  shell: |
    mysql --user=root <<EOF
    ALTER USER 'root'@'localhost' IDENTIFIED BY 'redhat';
    DELETE FROM mysql.user WHERE User='';
    DROP DATABASE IF EXISTS test;
    DELETE FROM mysql.db WHERE Db='test' OR Db='test\\_%';
    FLUSH PRIVILEGES;
    EOF
  args:
    executable: /bin/bash
  when: "'Access denied' not in mariadb_root_check.stderr"

# Create database and user
- name: Create database and user in MariaDB
  shell: |
    mysql --user=root --password=redhat <<EOF
    CREATE DATABASE vishnu_DB;
    CREATE USER 'vishnu'@'localhost' IDENTIFIED BY 'redhat';
    GRANT ALL PRIVILEGES ON vishnu_DB.* TO 'vishnu'@'localhost';
    FLUSH PRIVILEGES;
    EOF
  args:
    executable: /bin/bash

  register: db_creation_result
  failed_when: db_creation_result.rc != 0 and "exists" not in db_creation_result.stderr
  changed_when: "'CREATE DATABASE' in db_creation_result.stdout or 'GRANT' in db_creation_result.stdout"

- name: Copy wp-config.php
  template:
    src: /root/wp-config.php.j2 # Make sure this path is correct
    dest: /home/vishnu/website/public/wordpress/wp-config.php

- name: Navigate to the website public directory
  shell: cd /home/vishnu/website/public
  args:

```

```
args:
  chdir: /home/vishnu/website/public
  become: yes

- name: Download phpMyAdmin zip
  get_url:
    url: https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip
    dest: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages.zip

- name: Unzip phpMyAdmin zip
  unarchive:
    src: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages.zip
    dest: /home/vishnu/website/public/
    remote_src: yes

- name: Copy extracted phpMyAdmin to a new folder
  command: cp -r /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages /home/vishnu/website/public/phpMyAdmin

- name: Remove the extracted phpMyAdmin folder
  file:
    path: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages
    state: absent

- name: Remove the phpMyAdmin zip file
  file:
    path: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages.zip
    state: absent

- name: Add Match User vishnu block to /etc/ssh/sshd_config
  blockinfile:
    path: /etc/ssh/sshd_config
    block: |
      Match User vishnu
        ForceCommand internal-sftp
        PasswordAuthentication yes
    notify: Restart SSH service

- name: Change ownership of /home/vishnu
  command: chown root:root /home/vishnu
  become: yes

- name: Change ownership of /home/vishnu/website/public
  command: chown vishnu:vishnu /home/vishnu/website/public
  become: yes

- name: Restart SSH service
  service:
    name: sshd
```

```

service:
  name: sshd
  state: restarted

- name: Install Certbot and Nginx plugin
  yum:
    name:
      - certbot
      - python3-certbot-nginx
    state: present
    become: yes

- name: Configure Nginx server block for vishnu
  copy:
    dest: /etc/nginx/conf.d/myweb.conf
    content: |
      server {
        listen 80;
        listen [::]:80;
        server_name vishnu.rageshvk.site; # Default server name

        root /home/vishnu/website/public;

        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
      }
    become: yes

- name: Test Nginx configuration
  command: nginx -t
  register: nginx_test_result
  failed_when: "'syntax is okay' not in nginx_test_result.stderr and 'test is successful' not in nginx_test_result.stderr"
  become: yes

- name: Restart Nginx to apply changes
  systemd:
    name: nginx
    state: restarted
    become: yes

```

```

- name: Restart Nginx to apply changes
  systemd:
    name: nginx
    state: restarted
    become: yes

- name: Obtain SSL certificate using Certbot
  command: certbot --nginx -d vishnu.rageshvk.site --non-interactive --agree-tos --email vishnumahesh166@gmail.com
  register: certbot_result
  failed_when: certbot_result.rc != 0
  changed_when: "'Congratulations' in certbot_result.stdout or 'Certbot' in certbot_result.stderr"

- name: Reload Nginx after SSL setup
  systemd:
    name: nginx
    state: reloaded

```

- name: Set up MariaDB, PHP, Nginx, and configure website
hosts: localhost
become: yes
tasks:
 - name: Ensure /etc/yum.repos.d directory exists
file:
 - path: /etc/yum.repos.d
 - state: directory
 - name: Create MariaDB repository file
copy:
 - dest: /etc/yum.repos.d/MariaDB.repo
 - content: | [mariadb] name = MariaDB baseurl =
<http://yum.mariadb.org/10.6/rhel9-amd64> gpgkey =
<https://yum.mariadb.org/RPM-GPG-KEY-MariaDB>
gpgcheck = 1
 - name: Ensure python3-pip is installed
yum:
 - name: python3-pip
 - state: present
 - name: Install passlib library
pip:
 - name: passlib
 - state: present
 - name: Install PHP, MariaDB server, and required PHP extensions
yum:
 - name:
 - php - php-mysqlnd
 - php-mysqli
 - php-gd
 - php-mbstring
 - mariadb-server

- mariadb
 - state: present
- name: Install Nginx
 - yum:
 - name: nginx
 - state: present
- name: Start and enable MariaDB service
 - systemd:
 - name: mariadb
 - enabled: yes
 - state: started
- name: Start and enable Nginx service
 - systemd:
 - name: nginx
 - enabled: yes
 - state: started
- name: Create a user named vishnu
 - user:
 - name: vishnu
 - password: "{ { 'redhat' | password_hash('sha512') } }"
 - create_home: yes
- name: Create website directory for vishnu
 - file:
 - path: /home/vishnu/website/public
 - state: directory
 - owner: vishnu
 - group: vishnu
 - mode: '0755'
- name: Configure Nginx server block for vishnu
 - copy:
 - dest: /etc/nginx/conf.d/mydata.conf
 - content: |
 - server {
 - listen 80;

```

listen [::]:80;
server_name _; # Default server name
root /home/vishnu/website/public;
# Load configuration files for the default server block.
include /etc/nginx/default.d/*.conf;
error_page 404 /404.html;
location = /404.html {
    }
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    }
}

```

- name: Test Nginx configuration
 command: nginx -t
 register: nginx_test_result
 failed_when: "'syntax is okay' not in nginx_test_result.stderr and 'test is successful' not in nginx_test_result.stderr"

- name: Reload Nginx to apply changes
 systemd:
 name: nginx
 state: reloaded

- name: Set permissions for /home/vishnu
 command: chmod -R 755 /home/vishnu/
 become: yes

- name: Change ownership for /home/vishnu/website
 command: chown -R nginx:nginx /home/vishnu/website/
 become: yes

- name: Navigate to the website public directory
 shell: cd /home/vishnu/website/public
 args:

chdir: /home/vishnu/website/public
become: yes

- name: Download WordPress latest.zip
get_url:
url: <https://wordpress.org/latest.zip>
dest: /home/vishnu/website/public/latest.zip
- name: Unzip the WordPress archive
unarchive:
src: /home/vishnu/website/public/latest.zip
dest: /home/vishnu/website/public/
remote_src: yes
- name: Remove the downloaded zip file
file:
path: /home/vishnu/website/public/latest.zip
state: absent
- name: Set permissions for wp-content directory
command: chmod -R 755 /home/vishnu/website/public/wordpress/wp-content
become: yes
- name: Change ownership of wp-content directory
command: chown -R nginx:nginx
/home/vishnu/website/public/wordpress/wp-content
become: yes

Secure MariaDB installation using SQL commands, only if root password is not set

- name: Check if MariaDB root password is set
command: mysqladmin -u root status
register: mariadb_root_check

failed_when: false
changed_when: false

- name: Secure MariaDB installation

shell: |
mysql --user=root <<EOF
ALTER USER 'root'@'localhost' IDENTIFIED BY 'redhat';
DELETE FROM mysql.user WHERE User=";
DROP DATABASE IF EXISTS test;
DELETE FROM mysql.db WHERE Db='test' OR Db='test_%';
FLUSH PRIVILEGES;
EOF

args:

executable: /bin/bash

when: "'Access denied' not in mariadb_root_check.stderr"

#Create database and user

- name: Create database and user in MariaDB

shell: |
mysql --user=root --password=redhat <<EOF
CREATE DATABASE vishnu_DB;
CREATE USER 'vishnu'@'localhost' IDENTIFIED BY 'redhat';
GRANT ALL PRIVILEGES ON vishnu_DB.* TO
'vishnu'@'localhost'; FLUSH PRIVILEGES;
EOF

args:

executable: /bin/bash

register: db_creation_result

failed_when: db_creation_result.rc != 0 and "exists" not in
db_creation_result.stderr

changed_when: "'CREATE DATABASE' in db_creation_result.stdout or
'GRANT' in db_creation_result.stdout"

- name: Copy wp-config.php
template:
 - src: /root/wp-config.php.j2 # Make sure this path is correct
 - dest: /home/vishnu/website/public/wordpress/wp-config.php
- name: Navigate to the website public directory
shell: cd /home/vishnu/website/public
args:
 - chdir: /home/vishnu/website/publicbecome: yes
- name: Download phpMyAdmin zip
get_url:
 - url: <https://files.phpmyadmin.net/phpMyAdmin/5.2.1/phpMyAdmin-5.2.1-all-languages.zip>
 - dest: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages.zip
- name: Unzip phpMyAdmin zip
unarchive:
 - src: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages.zip
 - dest: /home/vishnu/website/public/
 - remote_src: yes
- name: Copy extracted phpMyAdmin to a new folder
command: cp -r /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages /home/vishnu/website/public/phpMyAdmin
- name: Remove the extracted phpMyAdmin folder
file:
 - path: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages
 - state: absent
- name: Remove the phpMyAdmin zip file

file:

path: /home/vishnu/website/public/phpMyAdmin-5.2.1-all-languages.zip
state: absent

- name: Add Match User vishnu block to /etc/ssh/sshd_config
blockinfile:

path: /etc/ssh/sshd_config
block: |
Match User vishnu
ForceCommand internal-sftp
PasswordAuthentication yes

notify: Restart SSH service

- name: Change ownership of /home/vishnu
command: chown root:root /home/vishnu
become: yes
- name: Change ownership of /home/vishnu/website/public
command: chown vishnu:vishnu /home/vishnu/website/public
become: yes
- name: Restart SSH service
service:
name: sshd
state: restarted
- name: Install Certbot and Nginx plugin
yum:
name:
 - certbot
 - python3-certbot-nginxstate: present
become: yes

- name: Configure Nginx server block for vishnu

copy:

dest: /etc/nginx/conf.d/myweb.conf

content: |

```
server {  
    listen 80;  
    listen [::]:80;  
    server_name vishnu.rageshvk.site; # Default server name  
    root /home/vishnu/website/public;  
    # Load configuration files for the default server block.  
    include /etc/nginx/default.d/*.conf;  
    error_page 404 /404.html;  
    location = /404.html {  
    }  
    error_page 500 502 503 504 /50x.html;  
    location = /50x.html {  
    }  
}
```

become: yes

- name: Test Nginx configuration

command: nginx -t

register: nginx_test_result

failed_when: "'syntax is okay' not in nginx_test_result.stderr and 'test is successful' not in nginx_test_result.stderr"

become: yes

- name: Restart Nginx to apply changes

systemd:

name: nginx

state: restarted

become: yes

- name: Obtain SSL certificate using Certbot
 command: certbot --nginx -d vishnu.rageshvk.site --non-interactive --agree-tos --email vishnumahesh166@gmail.com
 register: certbot_result
 failed_when: certbot_result.rc != 0
 changed_when: "'Congratulations' in certbot_result.stdout or 'Certbot' in certbot_result.stderr"

- name: Reload Nginx after SSL setup
 systemd:
 - name: nginx
 - state: reloaded

5. Run The Play-book

ansible-playbook ansible.yml -i inventory

```
[root@ip-192-168-1-87 ~]# ansible-playbook ansible.yml -i inventory
PLAY [Set up MariaDB, PHP, Nginx, and configure website] *****
TASK [Gathering Facts] *****
ok: [localhost]
TASK [Ensure /etc/yum.repos.d directory exists] *****
ok: [localhost]
TASK [Create MariaDB repository file] *****
ok: [localhost]
TASK [Ensure python3-pip is installed] *****
ok: [localhost]
TASK [Install passlib library] *****
ok: [localhost]
TASK [Install PHP, MariaDB server, and required PHP extensions] *****
ok: [localhost]
TASK [Install Nginx] *****
ok: [localhost]
TASK [Start and enable MariaDB service] *****
ok: [localhost]
TASK [Start and enable Nginx service] *****
ok: [localhost]
TASK [Create a user named vishnu] *****
changed: [localhost]
TASK [Create website directory for vishnu] *****
changed: [localhost]
TASK [Configure Nginx server block for vishnu] *****
ok: [localhost]
```

```
TASK [Configure Nginx server block for vishnu] *****
ok: [localhost]

TASK [Test Nginx configuration] *****
changed: [localhost]

TASK [Reload Nginx to apply changes] *****
changed: [localhost]

TASK [Set permissions for /home/vishnu] *****
changed: [localhost]

TASK [Change ownership for /home/vishnu/website] *****
changed: [localhost]

TASK [Navigate to the website public directory] *****
changed: [localhost]

TASK [Download WordPress latest.zip] *****
changed: [localhost]

TASK [Unzip the WordPress archive] *****
changed: [localhost]

TASK [Remove the downloaded zip file] *****
changed: [localhost]

TASK [Set permissions for wp-content directory] *****
changed: [localhost]

TASK [Change ownership of wp-content directory] *****
changed: [localhost]

TASK [Check if MariaDB root password is set] *****
ok: [localhost]
```

```
TASK [Secure MariaDB installation] *****
skipping: [localhost]

TASK [Create database and user in MariaDB] *****
ok: [localhost]

TASK [Copy wp-config.php] *****
ok: [localhost]

TASK [Navigate to the website public directory] *****
changed: [localhost]

TASK [Download phpMyAdmin zip] *****
changed: [localhost]

TASK [Unzip phpMyAdmin zip] *****
changed: [localhost]

TASK [Copy extracted phpMyAdmin to a new folder] *****
changed: [localhost]

TASK [Remove the extracted phpMyAdmin folder] *****
changed: [localhost]

TASK [Remove the phpMyAdmin zip file] *****
changed: [localhost]

TASK [Add Match User vishnu block to /etc/ssh/sshd_config] *****
ok: [localhost]

TASK [Change ownership of /home/vishnu] *****
changed: [localhost]

TASK [Change ownership of /home/vishnu/website/public] *****
changed: [localhost]
```

```
TASK [Restart SSH service] *****
changed: [localhost]

TASK [Install Certbot and Nginx plugin] *****
ok: [localhost]

TASK [Configure Nginx server block for vishnu] *****
changed: [localhost]

TASK [Test Nginx configuration] *****
changed: [localhost]

TASK [Restart Nginx to apply changes] *****
changed: [localhost]

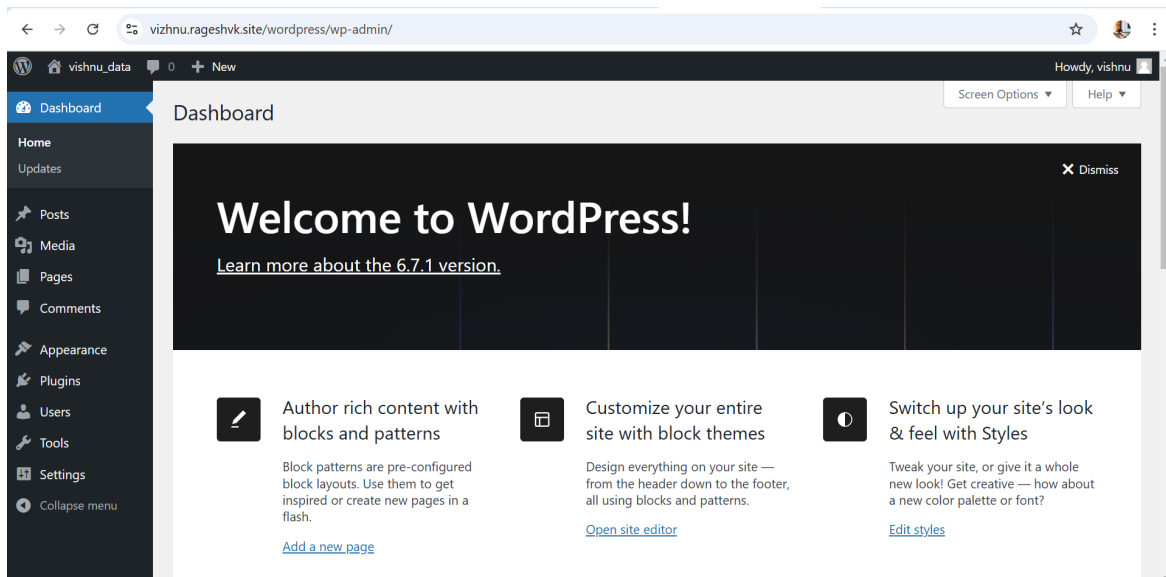
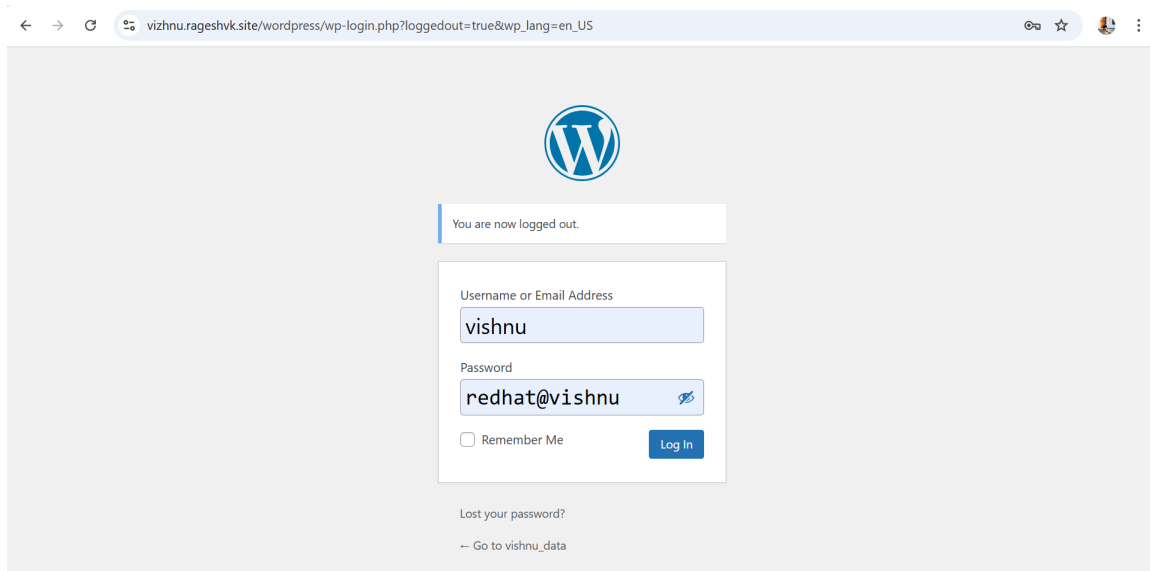
TASK [Obtain SSL certificate using Certbot] *****
changed: [localhost]

TASK [Reload Nginx after SSL setup] *****
changed: [localhost]

PLAY RECAP *****
localhost : ok=41 changed=25 unreachable=0 failed=0 skipped=1 rescued=0 ignored=0
```

6. Check the wordpress

Search: vizhnu.rageshvk.site/wordpress



7. Check PhpMyAdmin

Search: vizhnu.rageshvk.site/phpMyAdmin

The image shows two screenshots of the phpMyAdmin web interface. The top screenshot is the login page, and the bottom screenshot is the main dashboard.

Top Screenshot (Login Page):

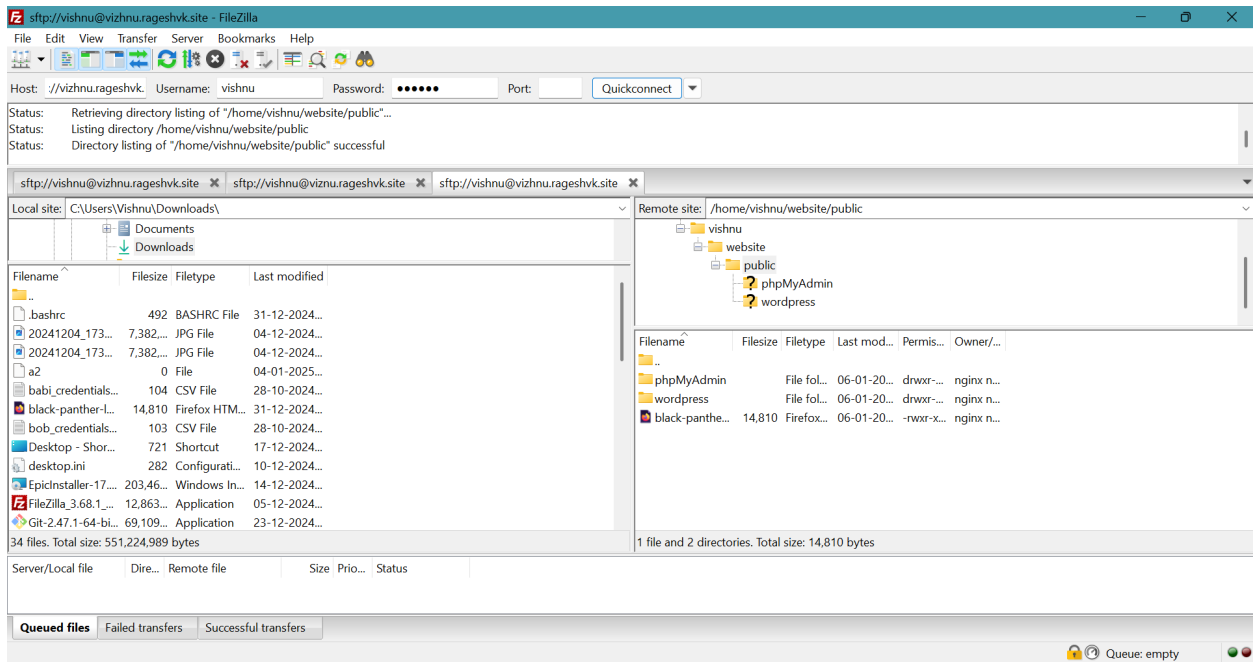
- URL: `vizhnu.rageshvk.site/phpMyAdmin/`
- Logo: phpMyAdmin with the text "Welcome to phpMyAdmin"
- Language: A dropdown menu set to "English".
- Log in: A button with a plus icon.
- Username: A text input field containing "vishnu".
- Password: A text input field with masked characters "*****".
- Log in: A button at the bottom right of the login form.

Bottom Screenshot (Dashboard):

- URL: `vizhnu.rageshvk.site/phpMyAdmin/index.php?route=/&route=%2F`
- Server: localhost
- Navigation tabs: Databases, SQL, Status, Export, Import, Settings, Variables,Charsets, Engines, Plugins.
- Left sidebar: Recent (New, information_schema, vishnu_DB) and Favorites.
- Main content area:
 - General settings:**
 - Change password: A link with a key icon.
 - Server connection collation: A dropdown menu set to "utf8mb4_unicode_ci".
 - More settings: A link with a wrench icon.
 - Appearance settings:**
 - Language: A dropdown menu set to "English".
 - Theme: A dropdown menu set to "pmahomme" with a "View all" button.
 - Database server:**
 - Server: Localhost via UNIX socket
 - Server type: MariaDB
 - Server connection: SSL is not being used
 - Server version: 10.6.20-MariaDB - MariaDB Server
 - Protocol version: 10
 - User: vishnu@localhost
 - Server charset: cp1252 West European (latin1)
 - Web server:**
 - nginx/1.26.2
 - Database client version: libmysql - mysqld 8.3.10
 - PHP extension: mysqli, curl, mbstring, sodium
 - PHP version: 8.3.10
 - phpMyAdmin:** A section at the bottom.
- Console: A tab at the bottom left.

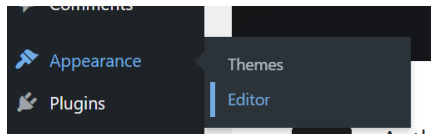
8. Check Filezillazilla

Host: **vizhnu.rageshvk.site** , username: **vishnu** , password: **redhat** , port: **22**



9. Create personal blog in wordpress

Navigate to wordpress > Appearance > Editor



– and create a blog > save

Then search on browser : vishnu.rageshvk.site/wordpress



Credentials:

- SFTP Credentials:

- **Host:** vishnu.rageshvk.site

- **Username:** [vishnu](#)

- **Password:** [redhat](#)

- phpMyAdmin Credentials:

- **URL:** vishnu.rageshvk.site/phpMyAdmin

- **Username:** [vishnu](#)

- **Password:** [redhat](#)

- WordPress Credentials:

- **WordPress Admin URL:** vishnu.rageshvk.site/wordpress/wp-admin

- **Username:** [vishnu](#)

- **Password:** [redhat@vishnu](#)

- **Personal Blog:** vishnu.rageshvk.site/wordpress