WordPress

"WordPress is a tool that helps you easily build websites. It gives you different designs to choose from, called 'themes', and lets you add extra features, called 'plugins'. You can edit your website's content without needing to know complicated code, and there's a big community of people who can help you if you need it. Overall, WordPress makes it simple to create your own corner of the internet, whether it's for sharing stories, selling things, or anything else you can imagine."

Key points about WordPress:

1) ***Open Source:*** *WordPress is open-source software, which means it is free to use, modify, and distribute. This has contributed to its widespread adoption and vibrant community.*

2) ***Ease of Use:*** *One of the main reasons for WordPress's popularity is its user-friendly interface, which allows users to create and manage websites without needing extensive technical knowledge.*

3) ***Themes****: WordPress offers a vast library of themes, which are pre-designed templates that determine the look and feel of a website. Users can choose from thousands of free and premium themes to customize their site's appearance.*

4) ***Plugins:*** *Plugins extend the functionality of WordPress websites. There are thousands of plugins available for various purposes, such as adding contact forms, optimizing SEO, integrating social media, and enhancing security.*

5) ***Flexibility:*** *WordPress is highly flexible and can be used to create different types of websites, including blogs, business websites, portfolios, forums, and e-commerce stores.*

6) ***SEO Friendly:*** *WordPress is inherently SEO friendly, with features that help optimize websites for search engines. Additionally, there are many SEO plugins available to further enhance optimization efforts.*

7) ***Community Support:*** *WordPress has a large and active community of users, developers, designers, and contributors who provide support, share knowledge, and create resources such as themes, plugins, and tutorials.*

8) ***Regular Updates:*** *The WordPress core software is regularly updated to improve security, performance, and functionality. Users are encouraged to keep their installations up to date to ensure they have the latest features and security patches.*

9) ***Growth Opportunities****: WordPress is not only a website platform but also a thriving ecosystem that provides opportunities for growth and monetization. Users can expand their websites with features like e-commerce, memberships, advertising, and affiliate marketing to generate revenue or achieve other goals.*

10) ***Security Measures:*** *WordPress offers various security measures to protect websites from vulnerabilities and malicious attacks. This includes features like user authentication, secure login, HTTPS support, regular security updates, and the availability of security plugins for additional protection.*

11) ***Global Reach:*** *WordPress powers millions of websites worldwide and supports content creation in multiple languages. It has a global impact and is used by individuals, businesses, organizations, and governments across different industries and sectors.*

12) ***Data Portability:*** *WordPress allows users to easily migrate their websites between different hosting providers or platforms. This ensures data portability and prevents vendor lock-in, giving users the freedom to switch hosts or platforms as needed.*

The platforms where we can deploy the WordPress website

1) ***WordPress.com*: WordPress.com is a hosted platform that offers a simplified way to create and deploy WordPress websites. Users can sign up for an account, choose a domain name, select a plan (which may include a free option with limitations or paid plans with additional features), and start building their website using WordPress's built-in tools. This option is ideal for beginners or users who prefer a hassle-free setup without worrying about server management.**

2) ***WordPress.org (Self-Hosted):* WordPress.org provides the open-source software that you can download and install on your own web hosting server. With a self-hosted WordPress site, you have full control over your website's customization, themes, plugins, and server environment. This option requires you to purchase a domain name and web hosting plan from a hosting provider such as Bluehost, SiteGround, or DreamHost. It offers maximum flexibility and scalability, making it suitable for businesses, developers, and users with specific requirements.**

3) ***Cloud Platforms:* Cloud platforms such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure provide infrastructure-as-a-service (IaaS) solutions that allow users to deploy WordPress on virtual machines (VMs) or containers. Users can configure and manage their server environments according to their requirements, but this option requires more technical expertise compared to managed hosting services.**

4) ***Managed WordPress Hosting:* Managed WordPress hosting providers offer specialized hosting services tailored specifically for WordPress websites. These providers optimize server configurations for WordPress, handle automatic updates, provide built-in security measures, and offer additional features such as staging environments and dedicated WordPress support. Examples of managed WordPress hosting providers include WP Engine, Kinsta, and Flywheel. Managed hosting is a good option for users who want high performance, reliability, and expert support without the hassle of server management.**

5) ***One-Click Installers:* Many web hosting providers offer one-click installers that allow users to quickly deploy WordPress with just a few clicks. These installers automate the process of downloading and installing WordPress on your hosting server, making it easy for beginners to get started. cPanel, Plesk, and Softaculous are examples of platforms that offer one-click WordPress installation.**

6) ***Local Development Environments:* Developers often use local development environments like XAMPP, MAMP, or Docker to set up WordPress sites on their computers for testing and development purposes before deploying them to live servers. This allows developers to work offline and experiment with customizations without affecting live websites.**

Practice: Hosting WordPress on AWS EC2 using Ubuntu with SSL and Domain Mapping

Hosting WordPress on AWS EC2 with SSL encryption and domain mapping using Ubuntu provides a secure and customizable solution for website deployment. This practice involves setting up an Ubuntu-based virtual server on AWS EC2, securing it with SSL encryption, and connecting a custom domain name to your WordPress site.

Step 1) Launch an Ubuntu-based AMI Ec2 Instance with SSH, HTTP, and HTTPS rules.
Step 2) Assigning an Elastic IP to the Ec2 Server is important in real-time scenarios.
Step 3) SSH to the Ec2 Instance.
Step 4) Run the following commands:

- ***sudo apt install apache2 -y*** → ***to install apache web server.***

After installing Apache, copy the Public IP of the Ec2 Instance and browse it to cross-check the apache web server is running.

- ***sudo apt install php libapache2-mod-php php-mysql -y*** → ***This command installs PHP along with necessary components for integrating PHP with the Apache web server and for connecting PHP with MySQL databases. This setup is common for hosting dynamic websites and web applications on servers running Apache and PHP. Once installed, PHP scripts can be executed by the web server, and PHP applications can interact with MySQL databases.***

- *sudo apt install mysql-server -y* → *to install MySQL*
- *sudo mysql -u root* → *to login MySQL server*

- *ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'Password@123';*
  → *Changes MySQL user 'root' password to 'Password@123' with native authentication.*

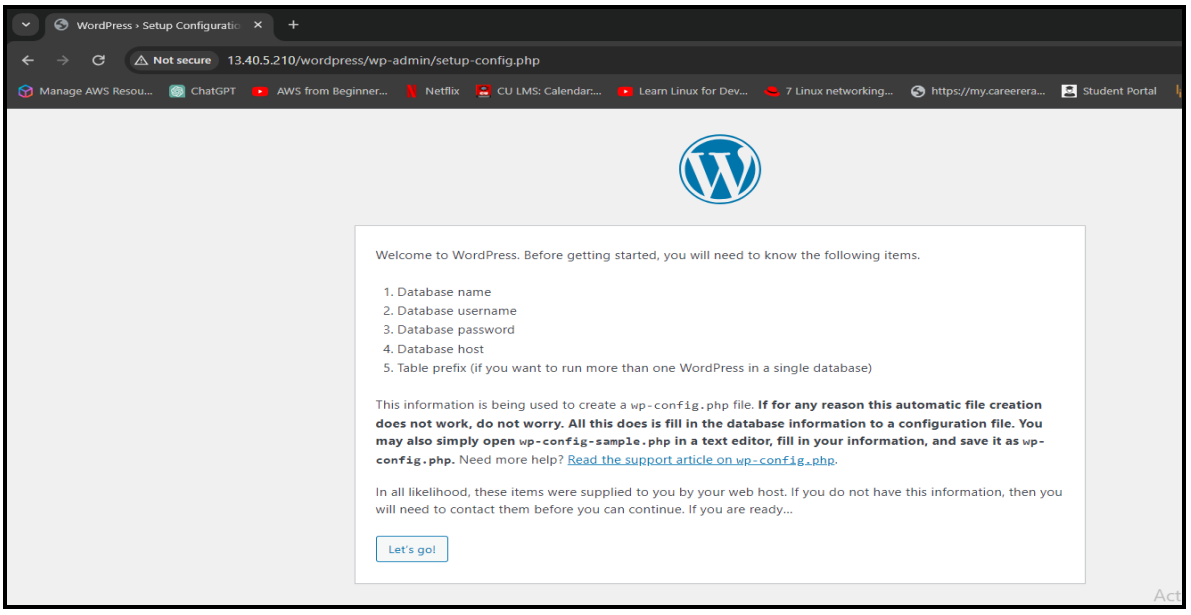
- *CREATE USER 'darsh'@localhost IDENTIFIED BY 'Password@123';* → *Creates a MySQL user named 'darsh' with the password 'Password@123', allowing access from 'localhost'.*

- *CREATE DATABASE wordpress;* → *Creates a new MySQL database named 'wordpress'.*

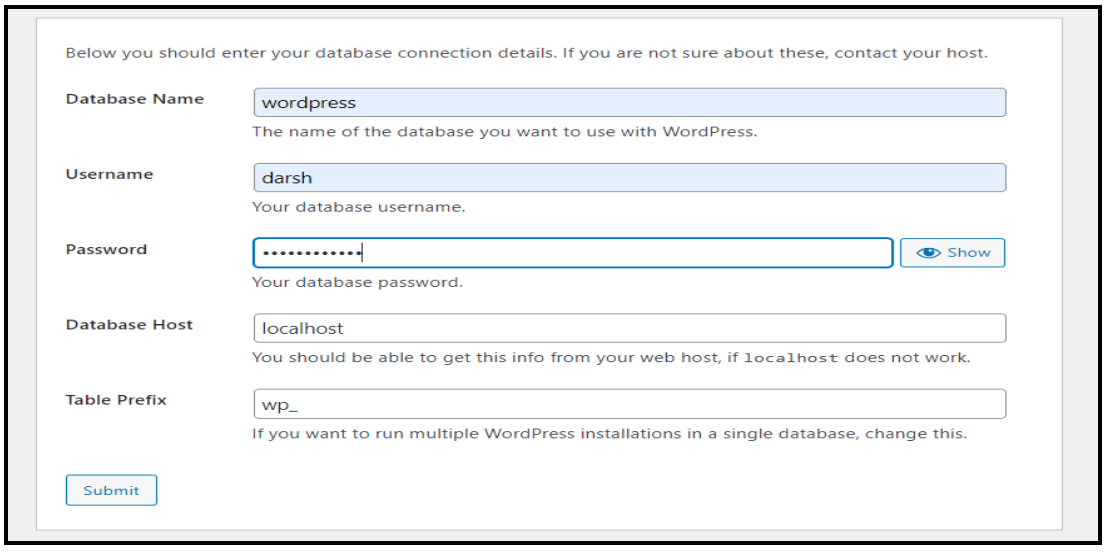
- *GRANT ALL PRIVILEGES ON wordpress.* TO 'darsh'@localhost;* → *Grants all privileges on the 'wordpress' database to the MySQL user 'darsh' when accessing from 'localhost'.*

- *wget https://wordpress.org/latest.tar.gz* → *Downloads the latest version of WordPress from the official website*
- *tar -xvf latest.tar.gz* → *Extracts the contents of the 'latest.tar.gz' file*
- *sudo mv wordpress/ /var/www/html* → *Moves the WordPress directory to the '/var/www/html' directory*
- *sudo systemctl restart apache2*

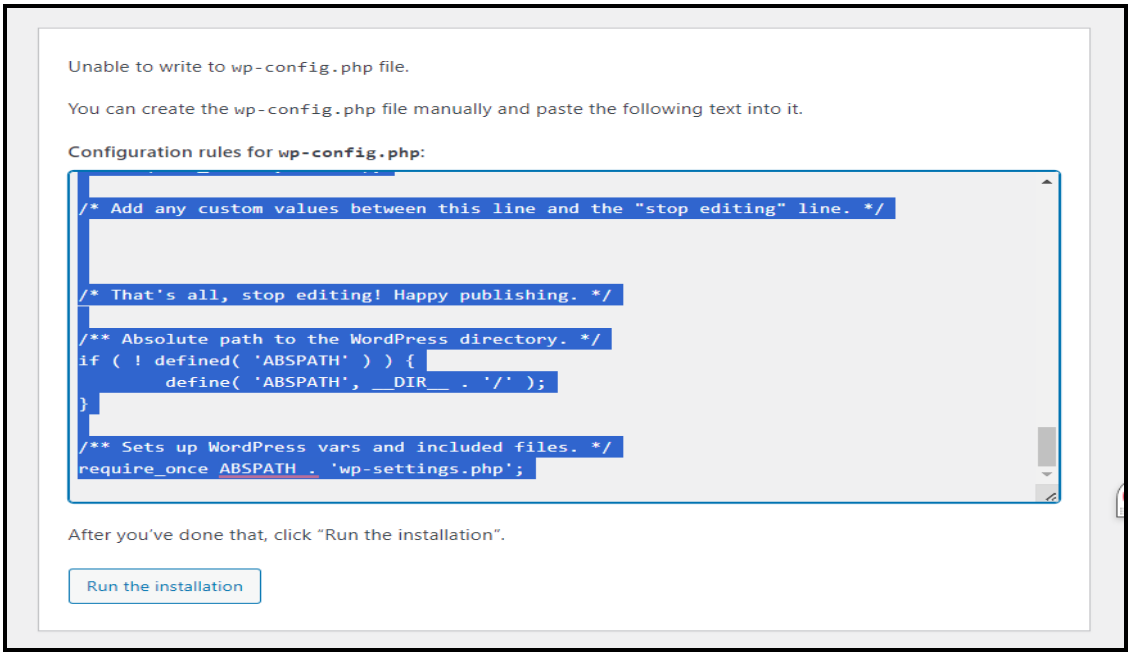
- ***Now browse the IP address by giving a path to /wordpress. A setup wizard will be open, click on the let's go button.***



- ***Enter the details shown in the below Image and click on the submit button.***



- ***Copy the configuration rule.***



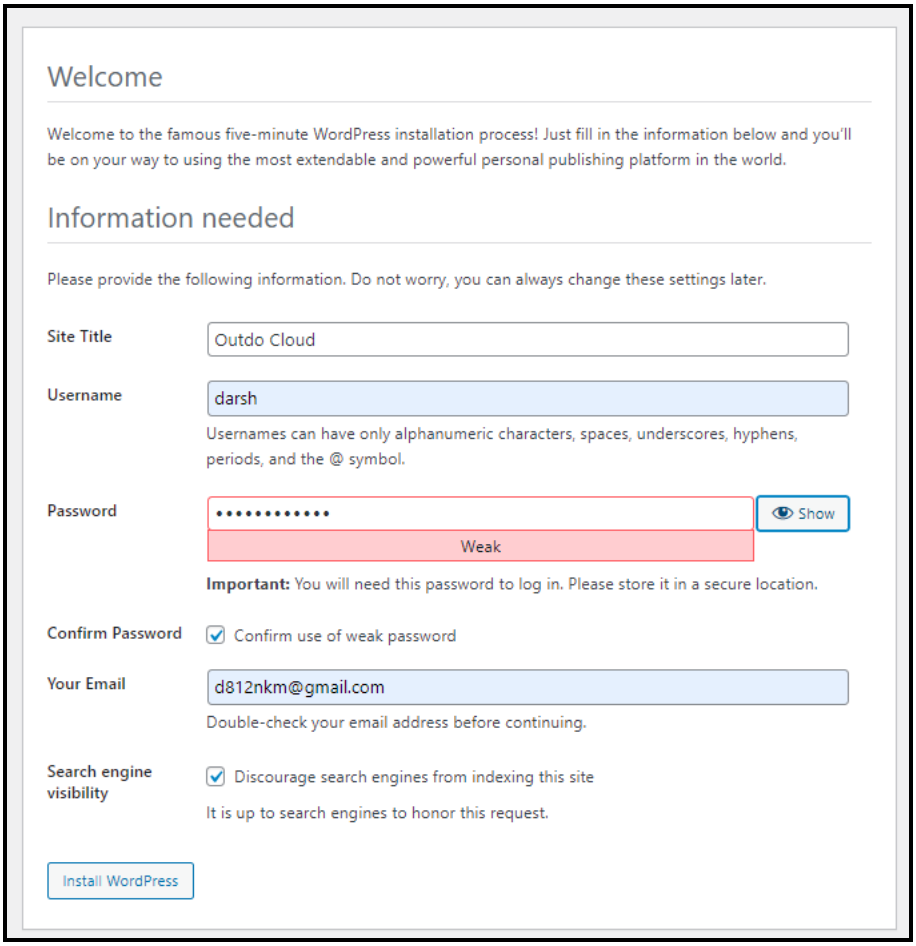
- ***Then we need to create wp-config.php, go back to the terminal, and run the command:***

➢ ***sudo vim /var/www/html/wordpress/wp-config.php***

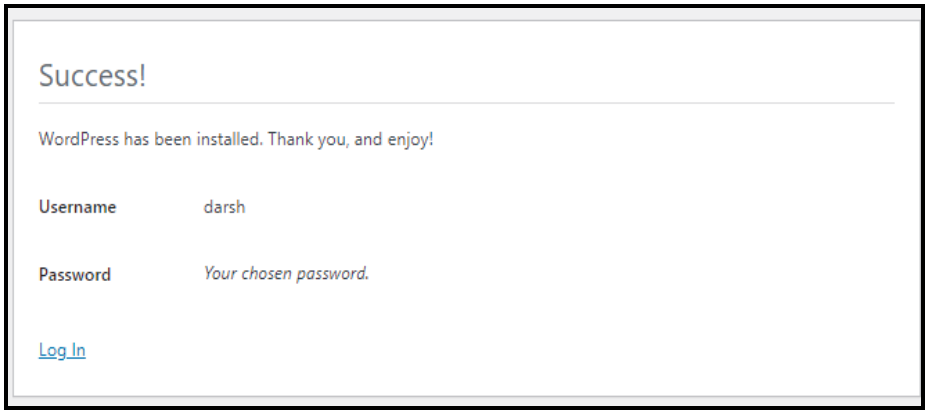
- ***Paste the copied configuration rule in the wp-config.php file, save, and exit from the file.***

- ***Now go to the WordPress setup wizard tab and click on the run installation button.***

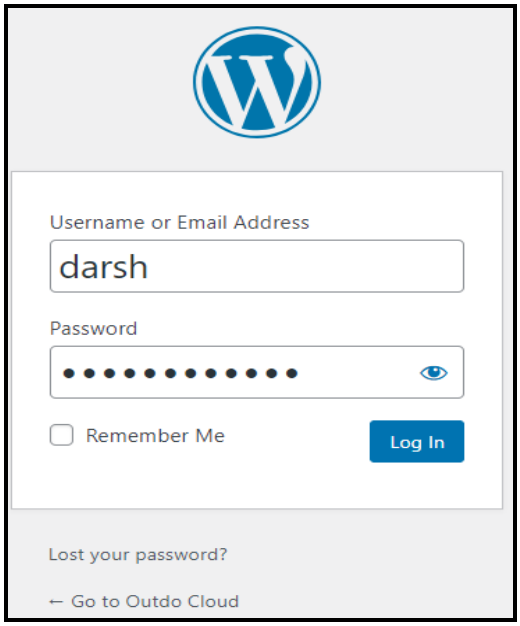
- ***WordPress install page will open, Enter the information shown in the below image, then click on the Install WordPress button.***



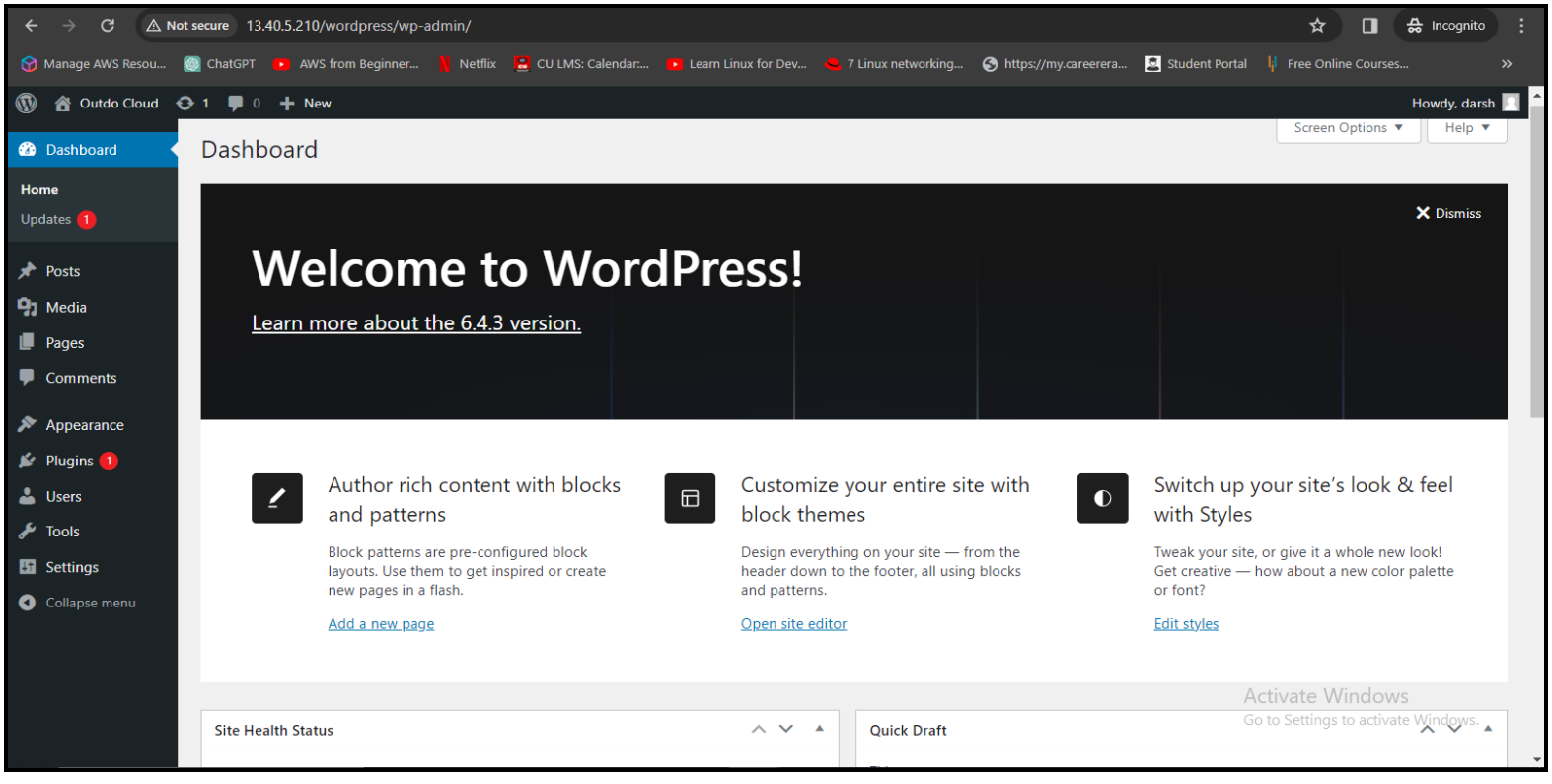
- ***We will receive a Success! Notification that WordPress has been installed.***



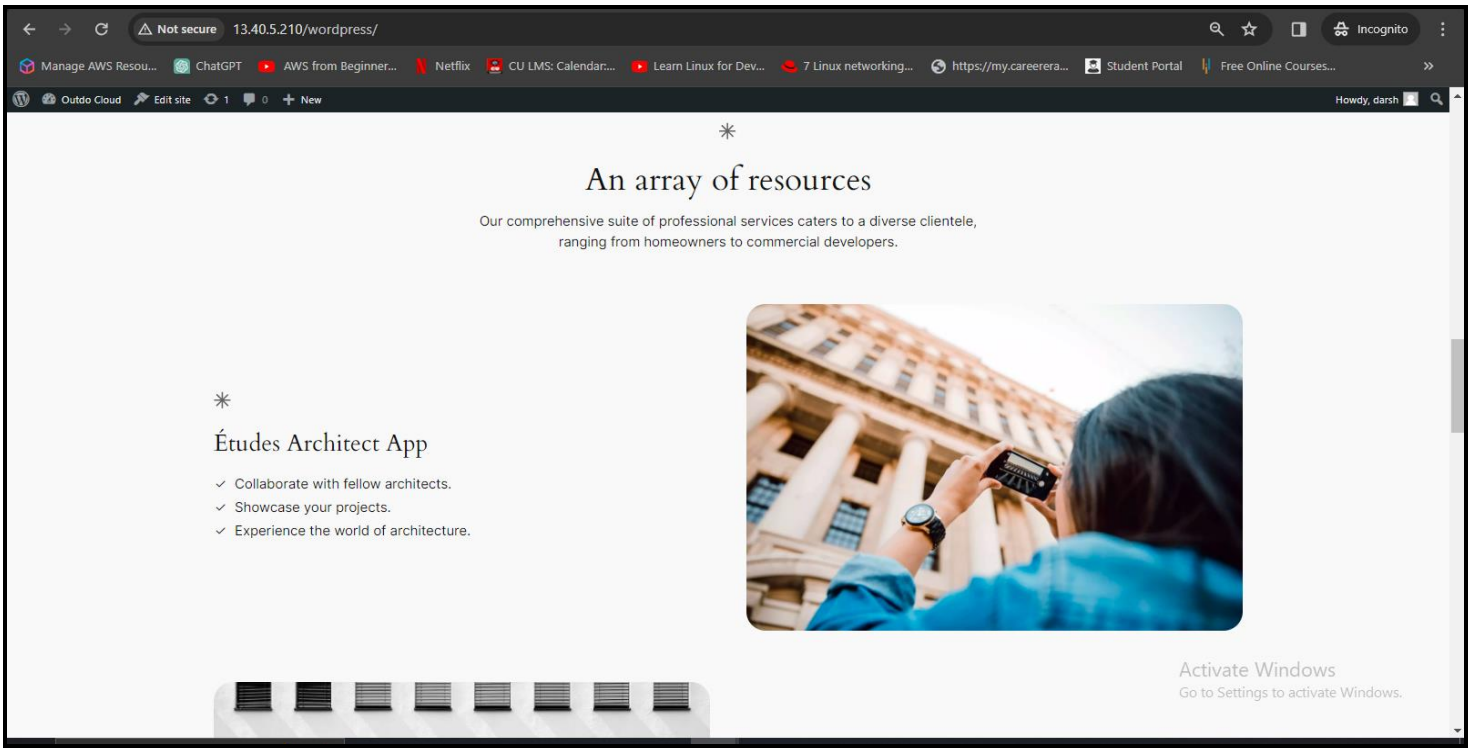
- ***Click on the login button, enter the username and password, and log in.***



- ***Now we have logged in WordPress dashboard here we can customize our WordPress Website.***



- ***Browse the IP Address with path /wordpress, default WordPress website will open.***



- ***We aim to ensure that when a customer or visitor lands on our website, they don't have to manually enter a path or IP address to access our website.***

- ***So here first we change in configuration, how Apache should handle website requests.***
  Way 1) Go back to the terminal and Run the following commands:
- *cd /etc/apache2/sites-available/*
- *ls*
- *sudo nano 000-default.conf*

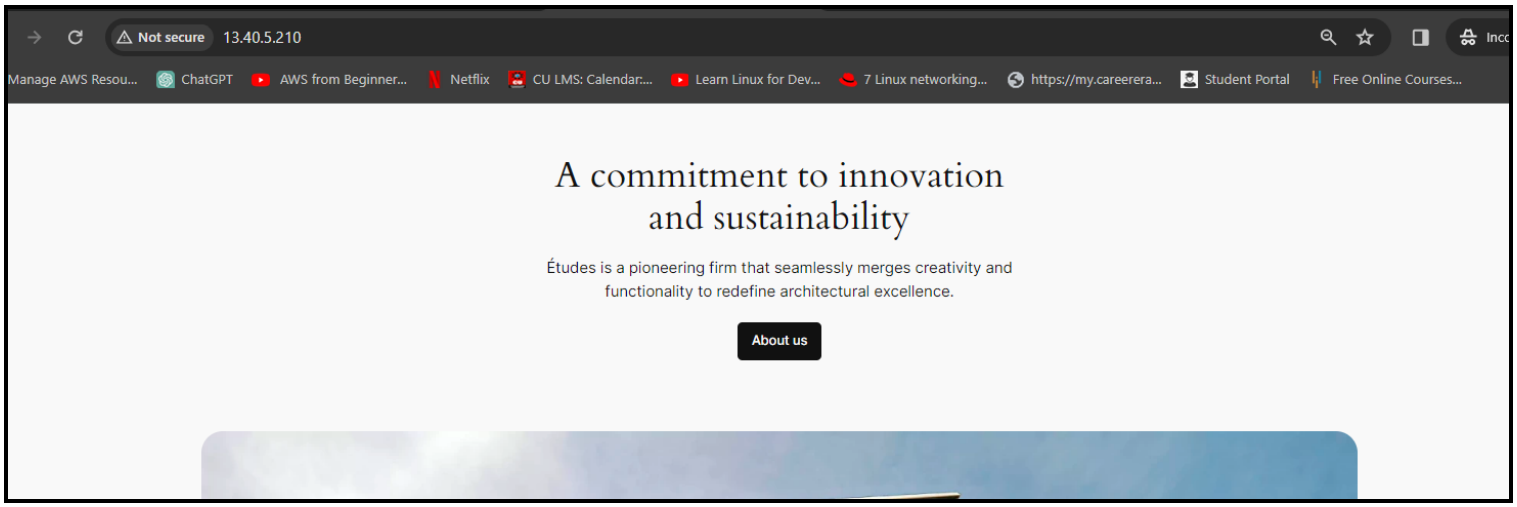
```
  GNU nano 6.2                                  000-default.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts,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

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

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

- ***Now in DocumentRoot add path /wordpress as shown in the image and save the file.***

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

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

- *sudo systemctl restart apache2*

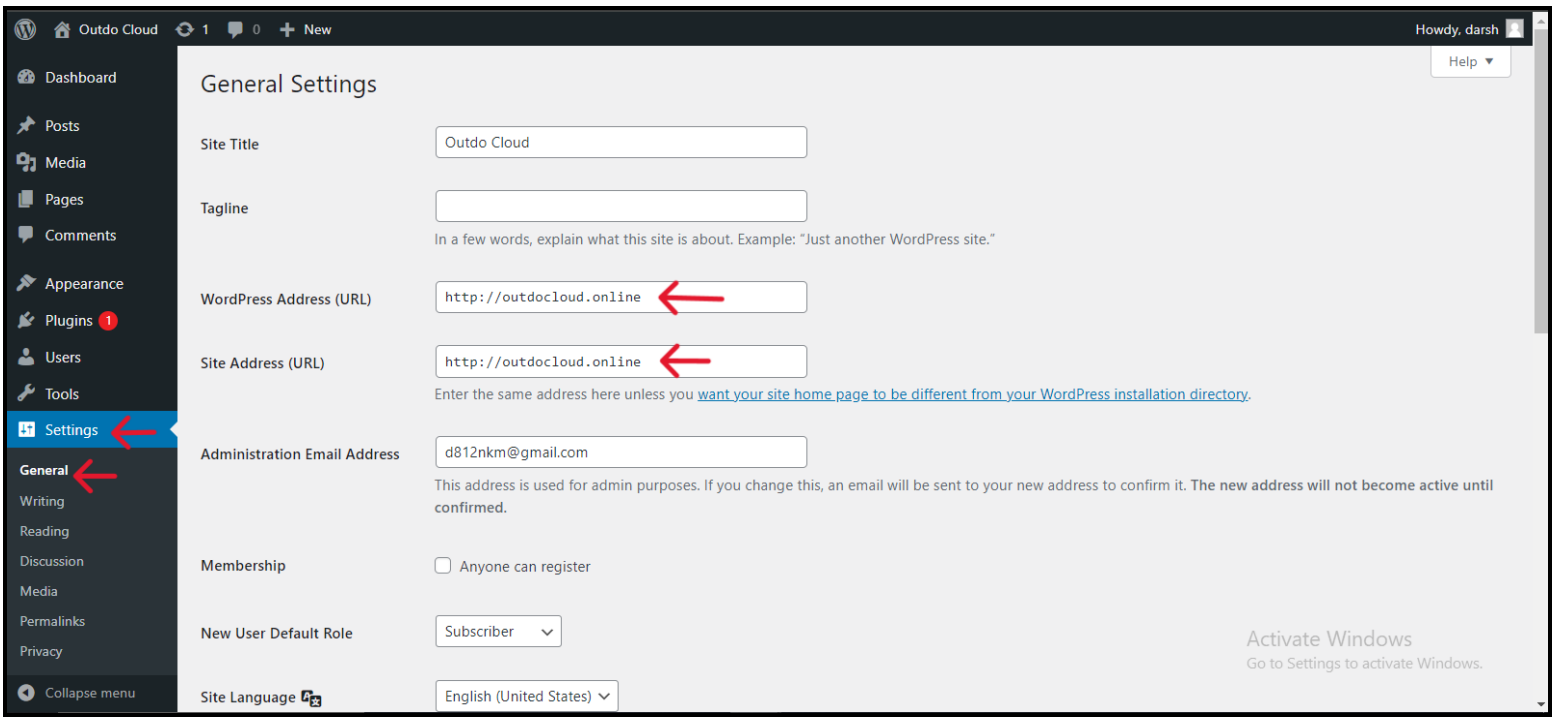
- ***Browse the IP Address***



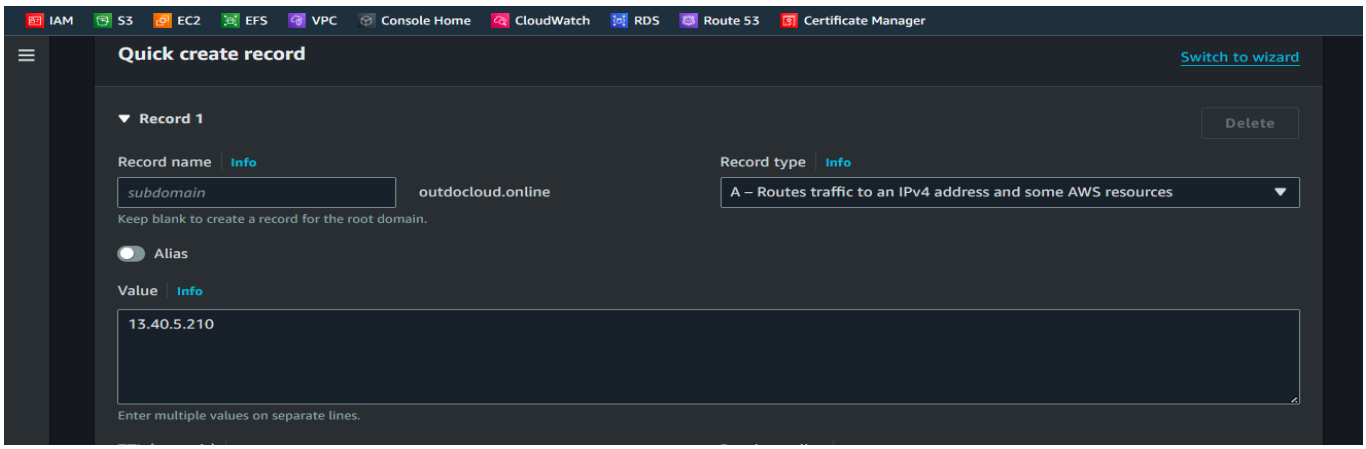
Here we can see now that only using the IP Address website is accessible.

Way 2) Using WordPress Dashboard.

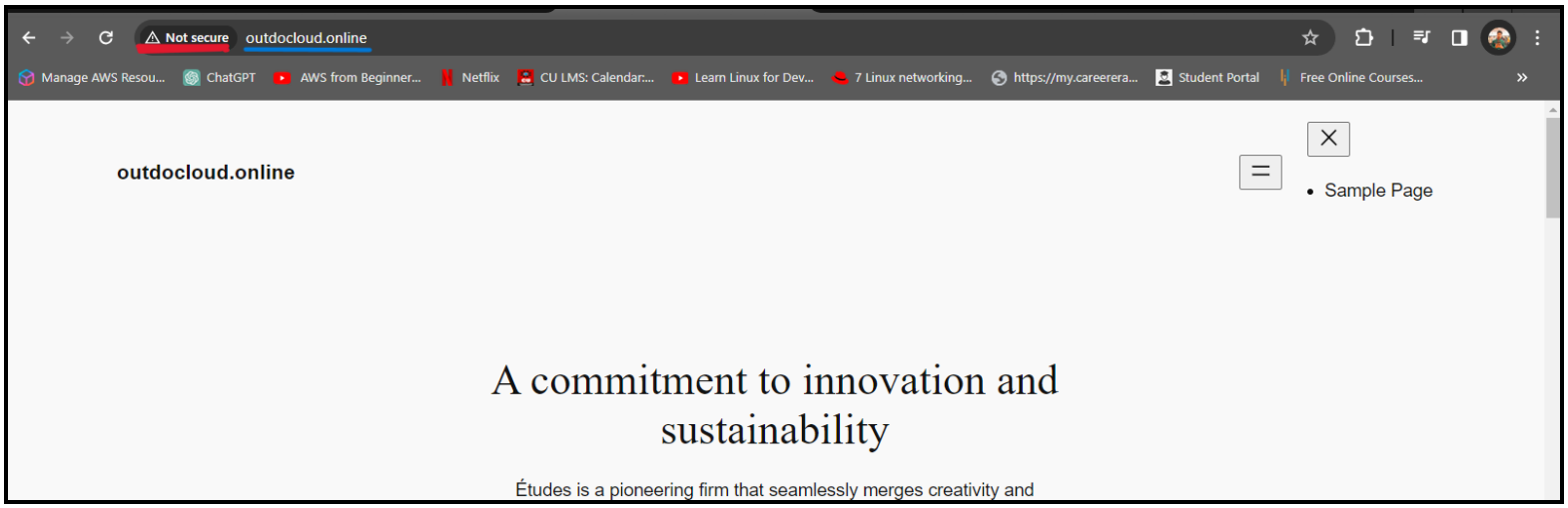
- ***Navigate to Settings and click on General.***
- ***In both URL boxes replace the IP Address with a Domain name, and save the changes.***



- ***Now we use our domain name to access our website***
- ***So, go to domain service and map this IP Address with DNS. Here I have a Hosted zone of my Domain in AWS Route 53, so I will create the A record.***



- ***Here we can see my WordPress website is now accessible using Domain Name.***



But here we can see our website is not secure.

Now we set up the SSL Certificate for our domain to secure our website using the Certbot tool.

Run the following Commands:

- ***sudo apt-get update***
- ***sudo apt install certbot python3-certbot-apache***
- ***sudo certbot –apache*** → ***After running this command Certbot Prompt will open, follow the process.***

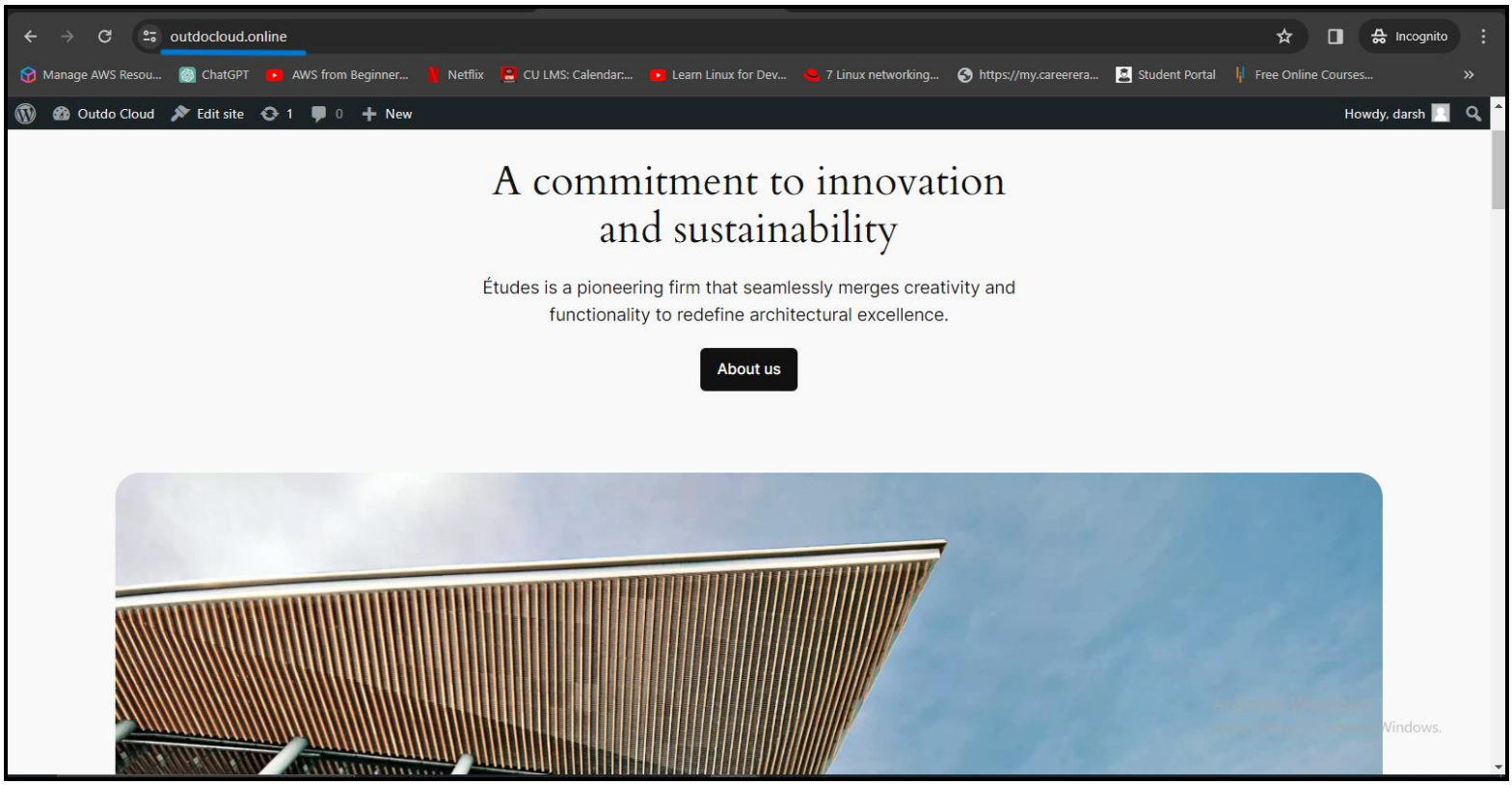
```
ubuntu@ip-172-31-28-87:~$ sudo certbot --apache
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Enter email address (used for urgent renewal and security notices)
 (Enter 'c' to cancel): d812nkm@gmail.com
```

- ***Enter the Email ID that is used during the WordPress Installation.***
- ***Enter your domain name.***

```
(Y)es/(N)o: y
Account registered.
Please enter the domain name(s) you would like on your certificate (comma and/or
space separated) (Enter 'c' to cancel): outdocloud.online
Requesting a certificate for outdocloud.online

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/outdocloud.online/fullchain.pem
Key is saved at:         /etc/letsencrypt/live/outdocloud.online/privkey.pem
This certificate expires on 2024-06-23.
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 outdocloud.online to /etc/apache2/sites-available/000-default-le-ssl.conf
Congratulations! You have successfully enabled HTTPS on https://outdocloud.online

If you like Certbot, please consider supporting our work by:
 * Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 * Donating to EFF:                    https://eff.org/donate-le
ubuntu@ip-172-31-28-87:~$
```

Here we have successfully received the SSL certificate for our domain outdocloud.online

Now browse Domain Name again

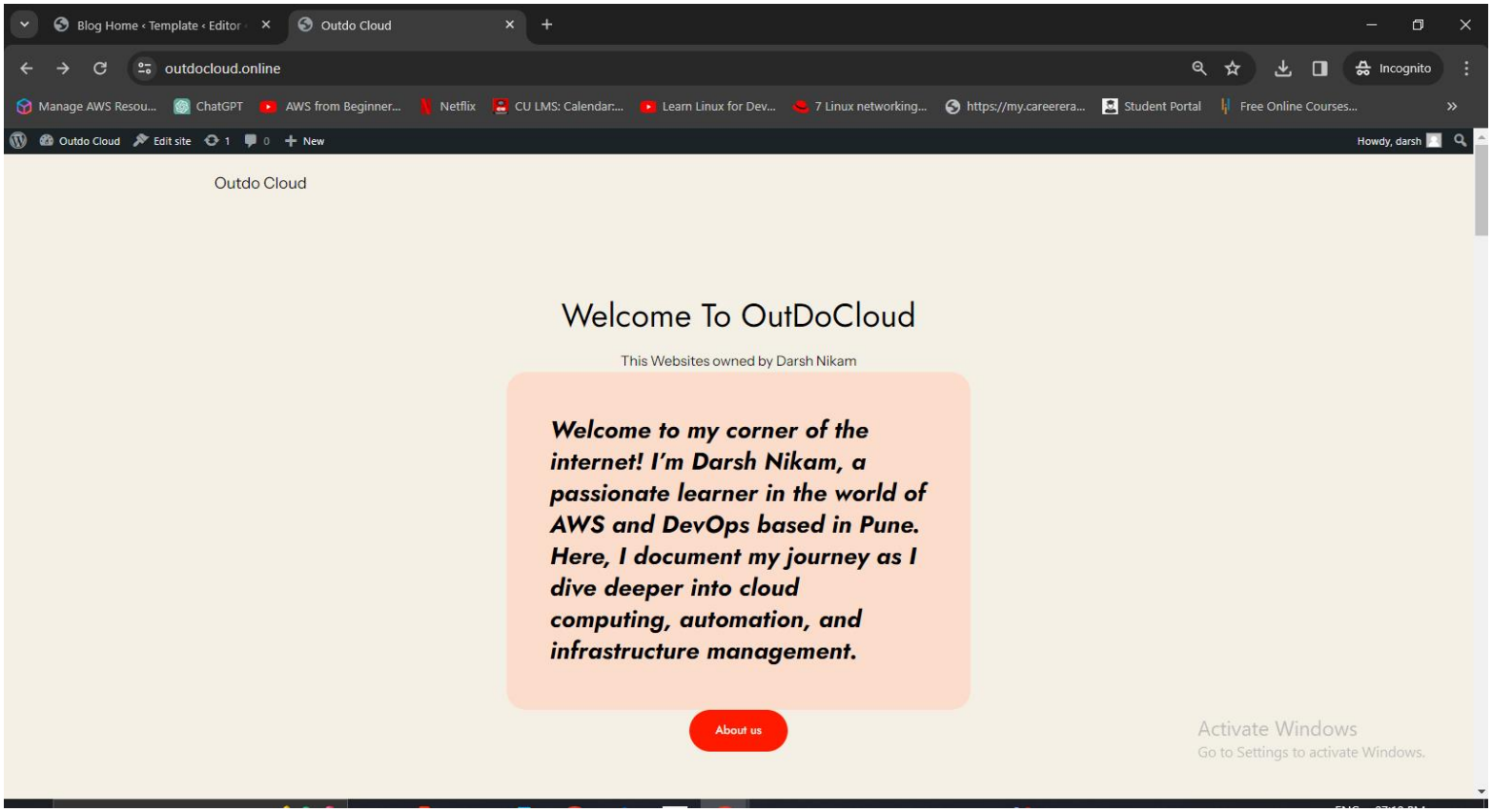


Here we can see now our website is secure.

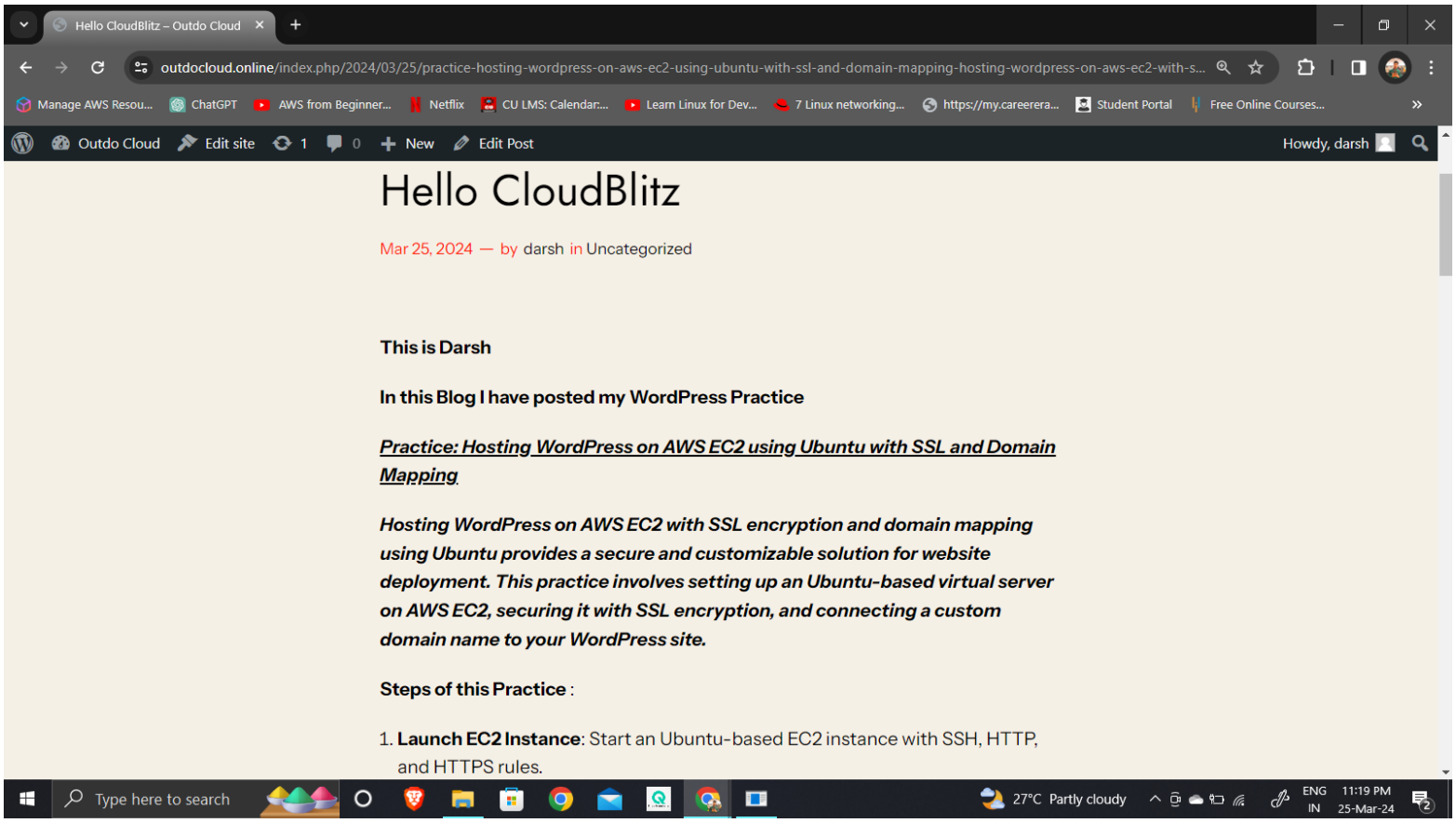
- ***To log into our WordPress Dashboard, navigate the domain.name/wp-admin***
  - ***outdocloud.online/wp-admin***

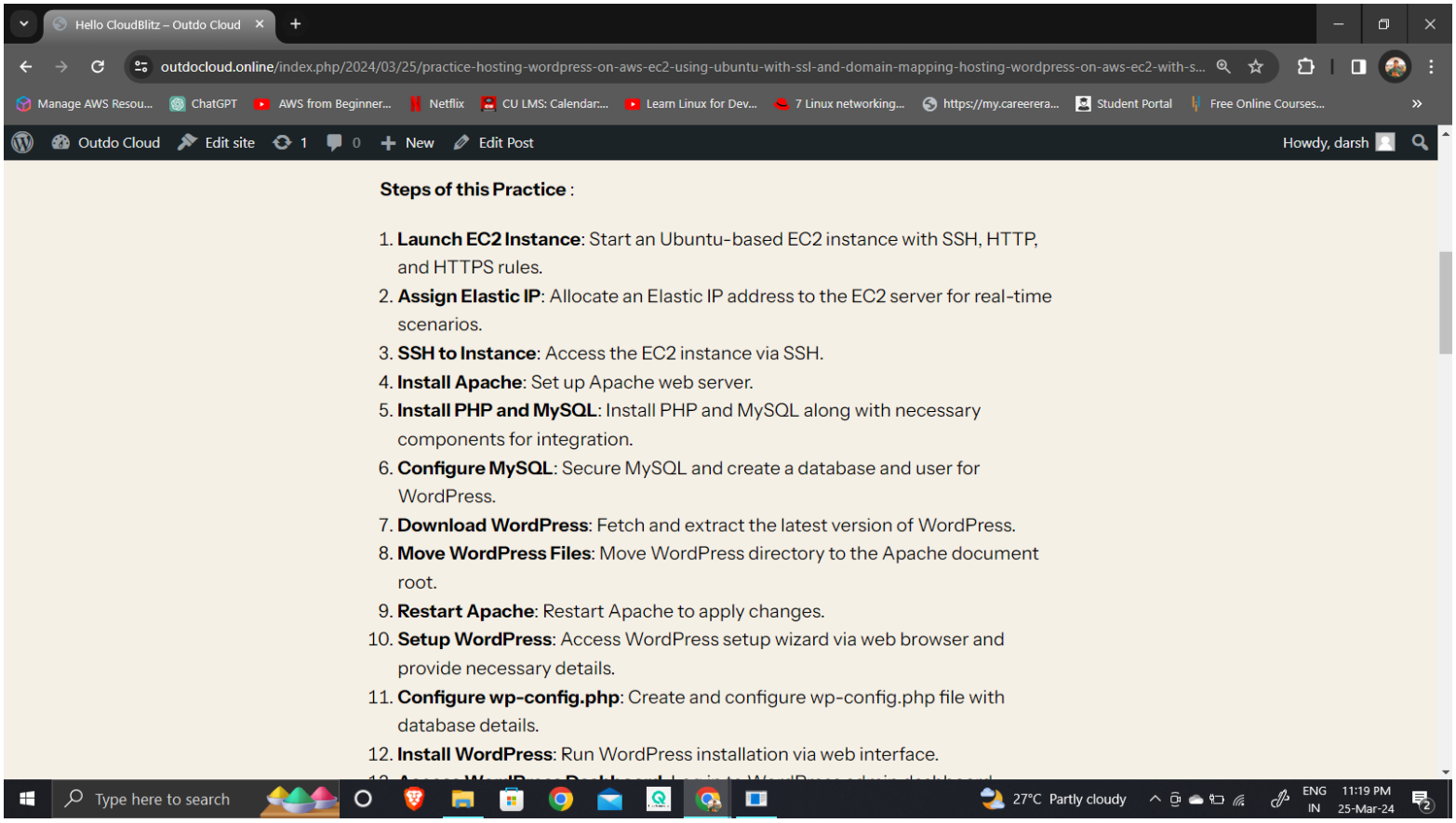
Enter username and password to log in. Here you can customize your website and post blogs.

My customized website



Blog posted on my website





13. **Access WordPress Dashboard**: Log in to WordPress admin dashboard.
14. **Configure Apache for Domain**: Adjust Apache configuration to handle website requests with the domain.
15. **Update WordPress Settings**: Update WordPress settings to use domain name instead of IP address.
16. **Map Domain to IP**: Configure DNS settings to map domain name to EC2 instance's IP address.
17. **Secure Website with SSL**: Install SSL certificate for domain using Certbot.
18. **Access WordPress Dashboard**: Log in to WordPress admin dashboard using the domain name.

Launch an Ubuntu-based AMI EC2 instance with SSH, HTTP, and HTTPS rules. Assign an Elastic IP for real-time accessibility. Configure the server by installing Apache, PHP, and MySQL. Set up WordPress by downloading, extracting, and moving files to the Apache document root. Configure MySQL, create a database and user for WordPress. Access the WordPress setup wizard, enter database and administrative details, and complete the installation. Configure Apache to handle website requests, directing to /wordpress. Alternatively, update WordPress settings to use a domain name for accessibility. Install an SSL certificate using Certbot for secure HTTPS access. Browse the domain name to verify secure access. Log in to the WordPress dashboard and customize the website.