Cloud Server Setup Documentation

Student Name: Astil Binu

Student Number: 35483676

IP Address: 16.170.77.192

DNS: https://techhoriozn.online/

1. Project Overview

1.1. Objective: The objective of this project is to create a cloud-based server using (IaaS) infrastructure as a service to host an Amazon EC2 server, which will provide information about different types of technology and other things with a technological background. The server will be configured and deployed manually, with all the required software and services installed from scratch.

1.2. Cloud Provider:

- Platform: Amazon EC2 Server
- Server Size: 2vCpus, 16GB Ram
- Operating system: Ubuntu

2. Environment Setup

2.1. Prerequisites:

- A Cloud account with Amazon EC2
- Access to SSH
- Basic knowledge of Linux commands and server administration.

2.2. Creating the Server Instance

- Log in to Amazon EC2 or any other cloud service provider according to your choice.

2.3. Create a New Server Instance:

Go to the Dashboard and click on Launch Instance

- Select a region, for example, I have chosen Europe, Stockholm
- Choose an Instance site, for example, choose the desired amount of RAM and storage and the CPUs.
- Select your preferred operating system, for example, Ubuntu.

- Set a Key pair name which you can easily access.
- Set up the Network, and while doing so, enable the allow HTTPS traffic from the internet.

2.4. Setup your Elastic IP Address:

- Go to Network and Security> Elastic IPs
- Click on Allocate Elastic Ip Address, and after that click on actions, and click on associate elastic ip address.
- From that, go to the instance and choose the desired instance to which you wish to allocate an Elastic Ip to it.
- Also allocate the private IP address to it which is already given.

3. Installing Apache On the Amazon EC2:

- Go the the command center and type in : **(ssh -i the location_of your_pem_file ec2-user@the ip address of your instance)**
- For example, mine is:
- **ssh -i D:\Looper.pem ubuntu@16.170.77.192**
- After gaining Connection to your instance then type in these commands:

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

3.1. Creating Your Website:

So first let's start by creating a Folder for the new website that you're going to create for that enter this:

```
sudo mkdir /var/www/gci/
```

I have named it gci here, but you can name it anything you want, and since now we have a directory created for our website, let's create an HTML file in it, and it can be created by typing this:

```
cd /var/www/gci/
nano index.html
```

Now you can edit the HTML file by using any HTML code you want, for example:

```
<html>
<head>
  <title> Ubuntu rocks! </title>
</head>
<body>
  <p> I'm running this website on an Ubuntu Server server!
</body>
</html>
```

3.2. Setting up a virtual host File:

- First, we must go into the configuration files directory for that type:

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

- Then type the following:

```
sudo cp 000-default.conf gci.conf
```

- Now let us edit the configuration file:

```
sudo nano gci.conf
```

- We also want the DocumentRoot directive to point to the directory where our site files are hosted:

```
DocumentRoot /var/www/gci/
```

- The default file doesn't come with a ServerName directive, so we'll have to add and define it by adding this line below the last directive:

```
ServerName gci.example.com
```

3.3. Activating the virtual host file:

- After setting up our website, we need to activate the virtual hosts configuration file to enable it. We do that by running the following command in the configuration file directory:

```
sudo a2ensite gci.conf
```

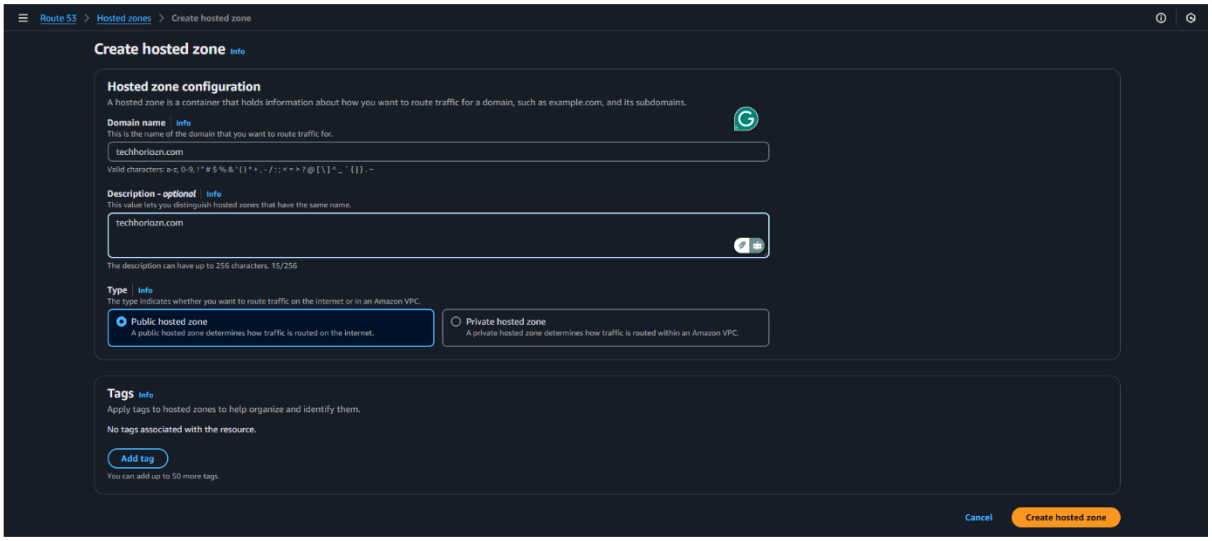
- You should see the following output

```
Enabling site gci.
To activate the new configuration, you need to run:
  service apache2 reload
root@ubuntu-server:/etc/apache2/sites-available#
```

- To load the new site, we restart Apache by typing:

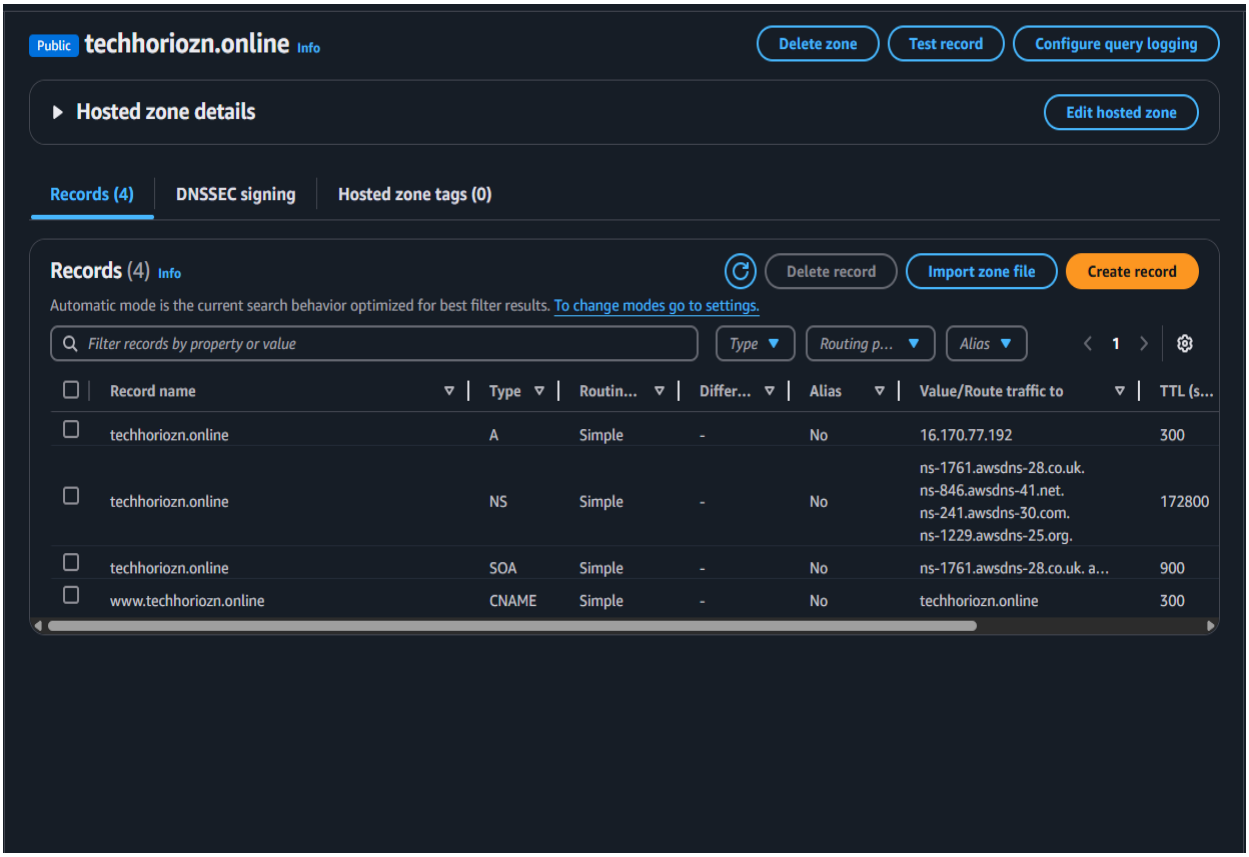
```
service apache2 reload
```

4. Global IP address and DNS configuration:
   4.1. First you have to purchase a domain name from any domain website for that I have used GoDaddy.
   4.2. After that, you have to register your domain name via GoDaddy/ route 53 in Amazon EC2. For that, first go to Route 53 and then click on Hosted Zones and then click Create Hosted Zone.



Like in the picture above, enter the details and click on Create Hosted Zones.

4.3. Now, after that, we have to create records like this, for example



5. **Let's install WordPress manually so that you can edit your Website:**

5.1. Go to Command Center and type in: **(ssh -i the location_of your_pem_file ec2-user@the ip address of your instance)**

5.2. Before installing nginx, we must disable Apache so first execute the following commands:

sudo systemctl stop apache2
sudo systemctl disable apache2

5.3. After that, let's install PHP, MariaDB and the Nginx Web Server for that type in this: **sudo apt install nginx mariadb-server php-fpm php-mysql**

5.4. Now, let's install WordPress: Execute the Following commands:
cd /var/www
sudo wget https://wordpress.org/latest.tar.gz
sudo tar -xzvf latest.tar.gz
sudo rm latest.tar.gz
sudo chown -R www-data:www-data wordpress
sudo find wordpress/ -type d -exec chmod 755 {} \;
sudo find wordpress/ -type f -exec chmod 644 {} \;

5.5. Let's Set Up the Database: Secure your MariaDB installation by adding a password and disabling other features. When prompted, answer Y. So now let's execute this command:

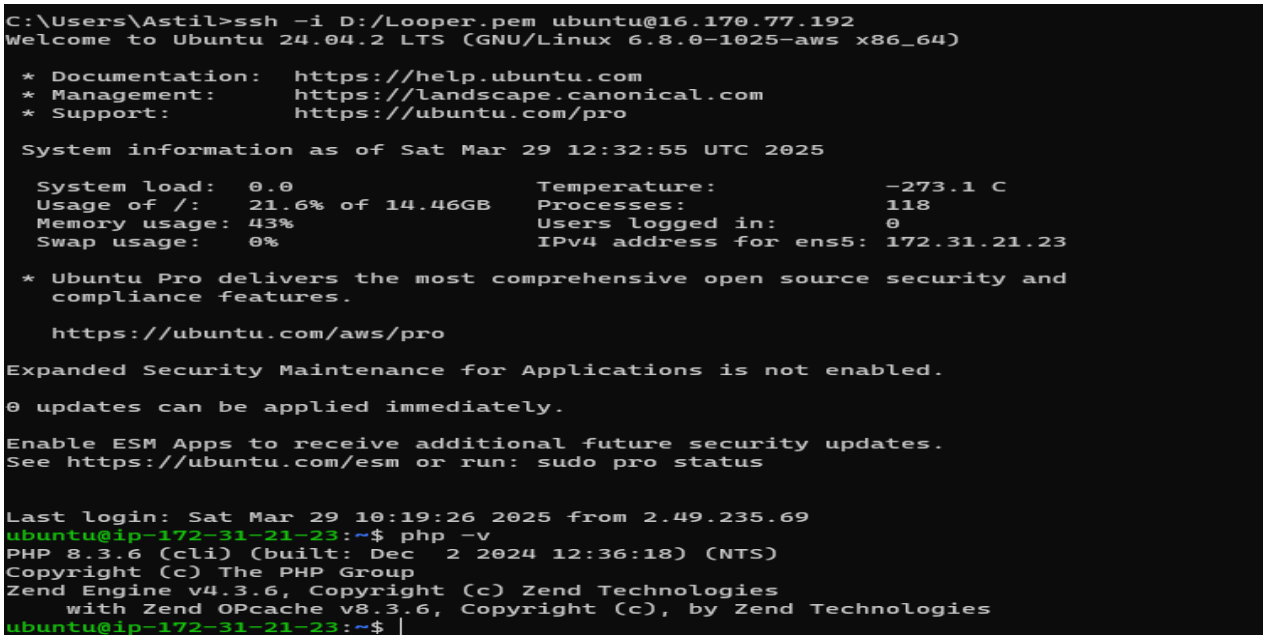
sudo mysql_secure_installation

- Now let's access the MariaDB console for that type in this:
Sudo mysql -u root -p

- Now, within the MariaDB console, let's create a database for WordPress and make sure to choose your database, username and password. So let’s execute this example Command:
create database example_db default character set utf8 collate utf8_unicode_ci;
create user 'example_user'@'localhost' identified by 'example_pw';
Grant all privileges on example_db.* TO 'example_user'@'localhost';
flush privileges;
exit

5.6. After this run this command to know which PHP version you have installed: **php -v**

And as you see, I have PHP version **8.3.6**.



5.7. Now, Let's Configure The Nginx Web Server:

- First go to the directory which contains the configuration files for the Nginx web server and create a new configuration file. To execute this command:

  Cd/etc/nginx/sites-available/

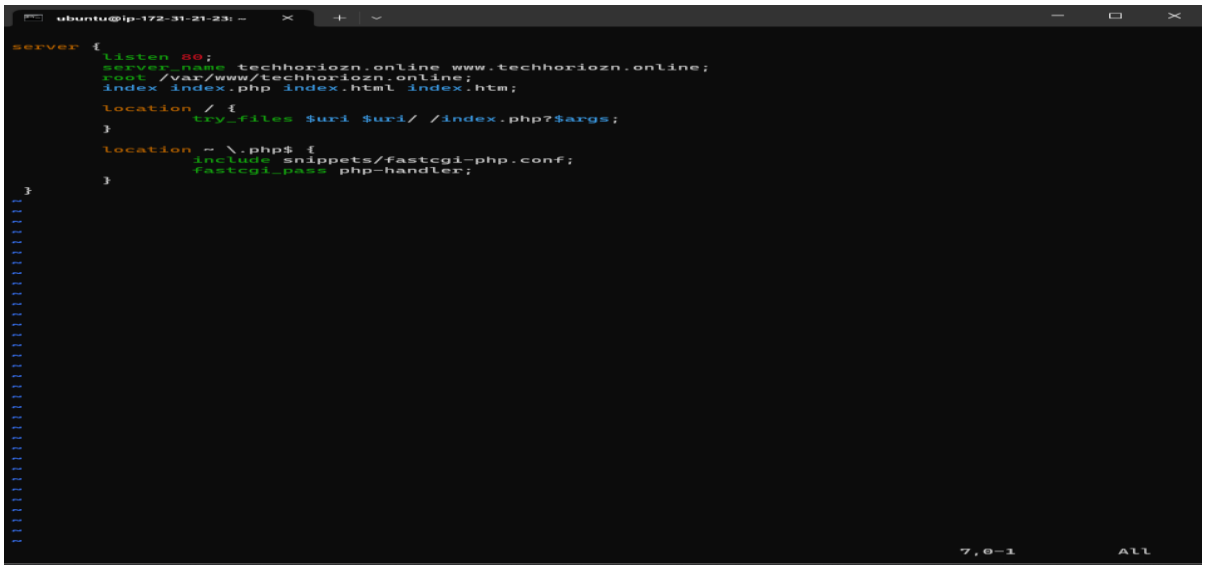
  sudo vim wordpress.conf

- Now use this configuration as an example and change it according to your website and server name. To input your command, press I and then enter your command, for example :

```
server {
        listen 80;
        server_name techhoriozn.online www.techhoriozn.online;
        root /var/www/techhoriozn.online;
        index index.php index.html index.htm;

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

```
location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass php-handler;
}
```

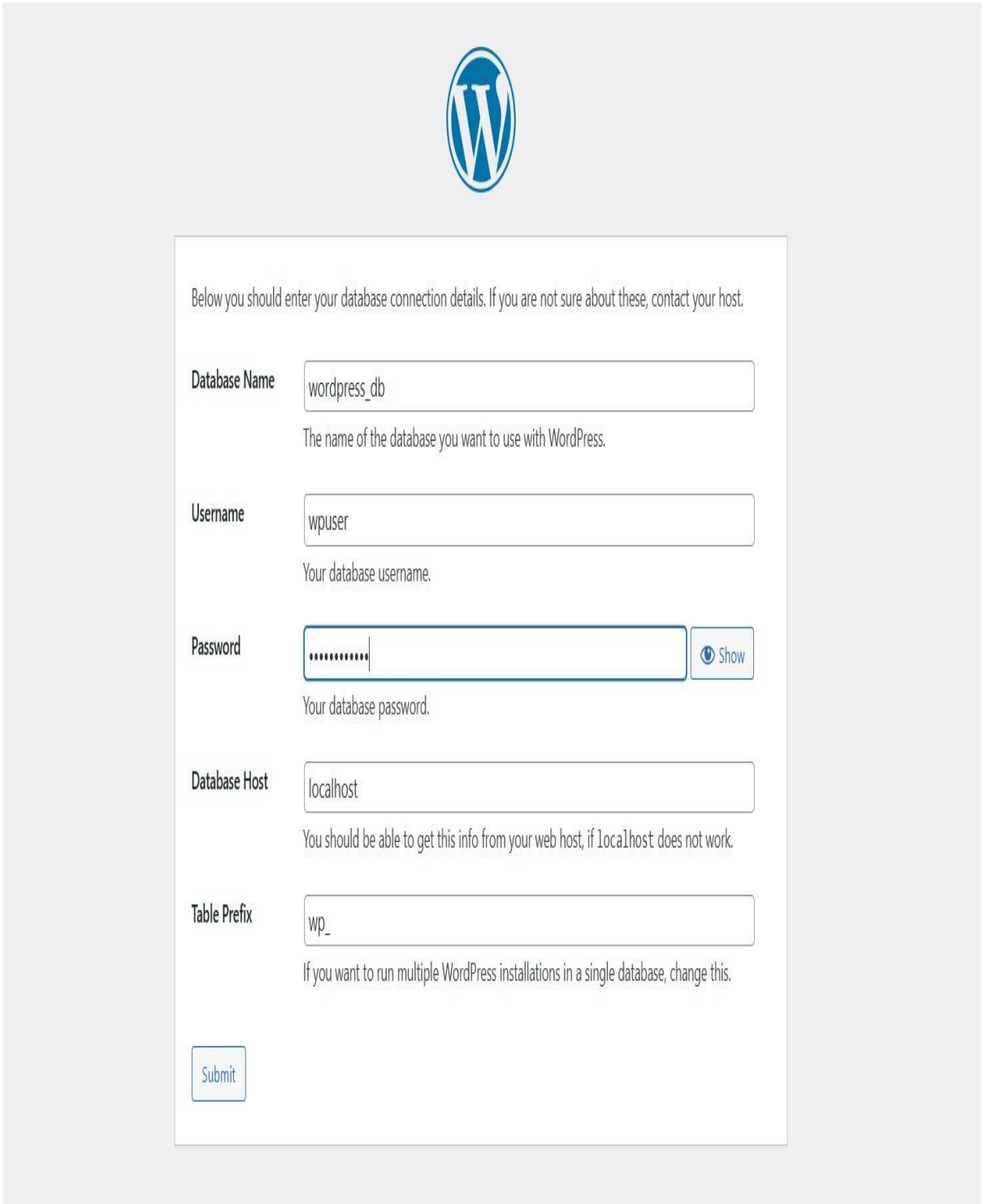


- Now, after this, press ESC to save your file, and then shift+; and then enter WQ exit so that it gets written, and you quit.

- After this, make a symbolic link to the website to let the Nginx know, and you can apply the changes by restarting the web server. To execute these commands:
  sudo ln -s /etc/nginx/sites-available/wordpress.conf /etc/nginx/sites-enabled/
  sudo nginx -t
  sudo systemctl restart nginx

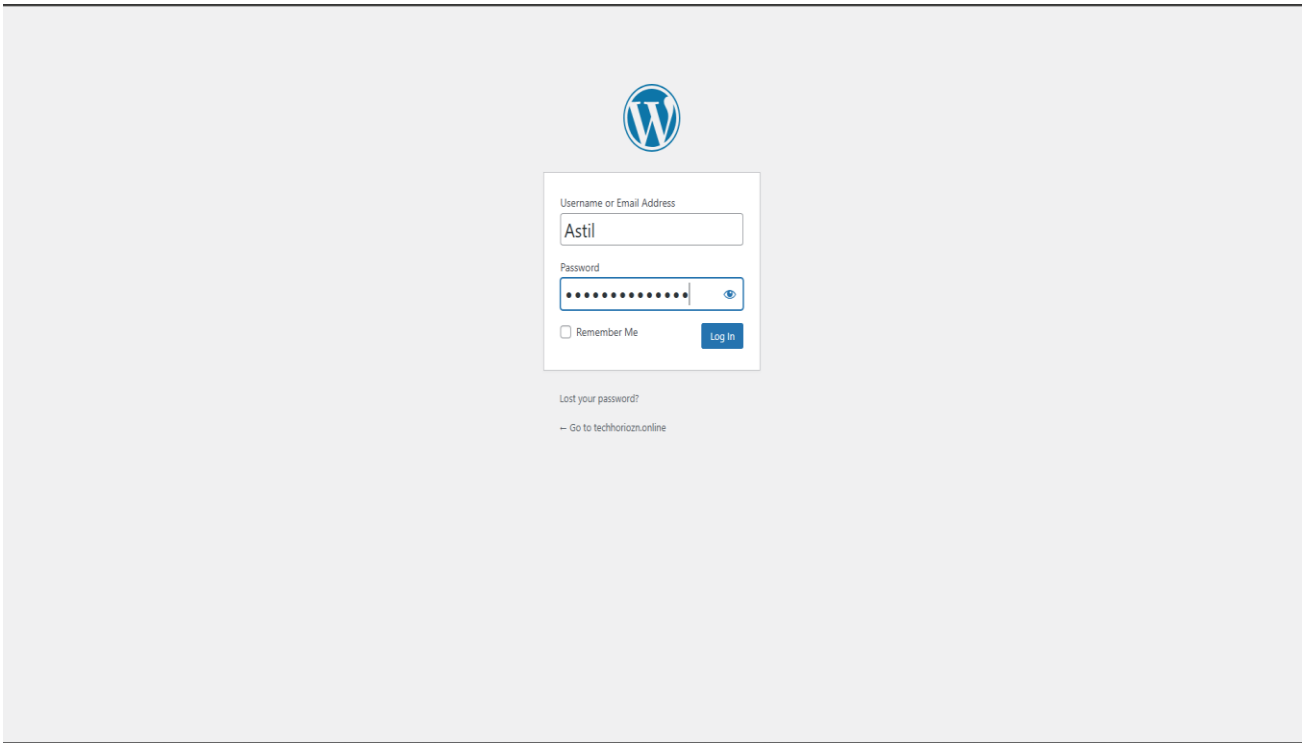
5.8. Finish the WordPress Install:

- Go to your domain in the browser and then enter your database name, user and password that u created earlier, like this.



After the installation, you can access your WordPress administrator dashboard at http://techhoriozn.com/wp-admin/, where example.com is your domain name.

It will look like this:



5.9. The only thing left is to install the PHP packages required by WordPress to go back to the command center, and enter this:

sudo apt install php-curl php-dom php-mbstring php-imagick php-zip php-gd

6. Installing an SSL certificate for HTTPS:
   For that, just execute the following command in your command center:
   sudo apt install certbot python3-certbot-nginxsudo certbot --nginx -d techhoriozn.online -d www.techhoriozn.online