**To Install Linux, Apache, MySQL, PHP (LAMP) Stack on Ubuntu**

<https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu>

<https://www.cherryservers.com/blog/install-lamp-on-ubuntu-22-04>

<https://cloudinfrastructureservices.co.uk/how-to-install-wordpress-on-ubuntu-20-04/>

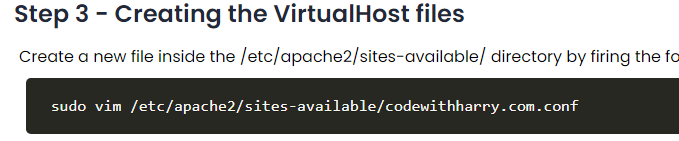
<https://www.codewithharry.com/blogpost/install-phpmyadmin-ubuntu/>

**Step — Configure & creating multiple virtual host**

<https://www.codewithharry.com/blogpost/host-multiple-websites-ubuntu-vps/>

sudo mkdir /var/www/your\_domain

sudo chown -R $USER:$USER /var/www/your\_domain



sudo nano /etc/apache2/sites-available/your\_domain.conf

This will create a new blank file. Add in the following bare-bones configuration with your own domain name:

<VirtualHost \*:80>

ServerName your\_domain

ServerAlias www.your\_domain

ServerAdmin webmaster@localhost

DocumentRoot /var/www/your\_domain

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Save and close the file when you’re done. If you’re using nano, do that by pressing CTRL+X, then Y and ENTER.

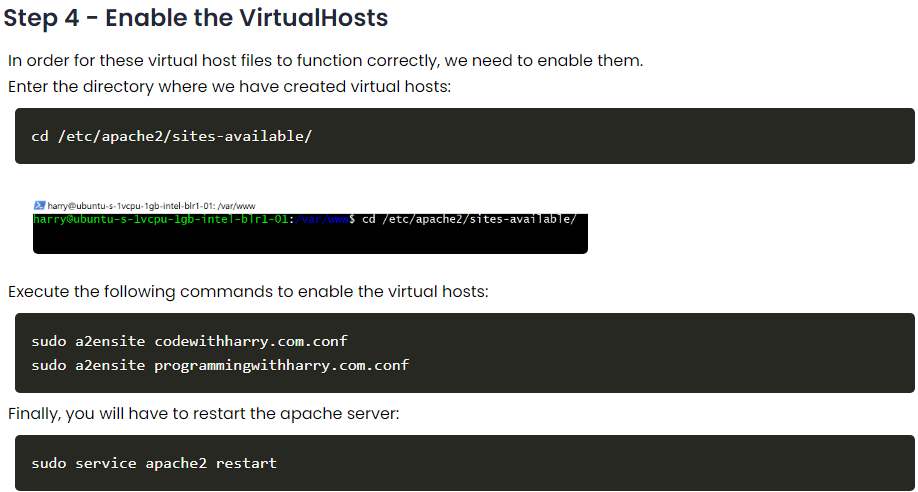
sudo a2ensite your\_domain

You might want to disable the default website that comes installed with Apache. This is required if you’re not using a custom domain name, because in this case Apache’s default configuration would override your virtual host. To disable Apache’s default website, type:

sudo a2dissite 000-default

To make sure your configuration file doesn’t contain syntax errors, run the following command:

sudo apache2ctl configtest



Finally, reload Apache so these changes take effect:

sudo systemctl reload apache2

nano /var/www/your\_domain/index.html

sudo nano /etc/apache2/mods-enabled/dir.conf

<IfModule mod\_dir.c>

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

sudo systemctl reload apache2

Server Ip Pointing on Domain name



## [Step — Installing Apache and Updating the Firewall](https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu#step-1-installing-apache-and-updating-the-firewall)

sudo -i          for changing root user

sudo chmod -R  777 /var/www/html

sudo apt update

sudo apt install apache2 –y

sudo systemctl status apache2

sudo apt update

sudo apt install apache2

or

sudo apt install apache2 -y

sudo systemctl status apache2

sudo ufw status     //should be inactive

//if active

sudo ufw allow 80/tcp

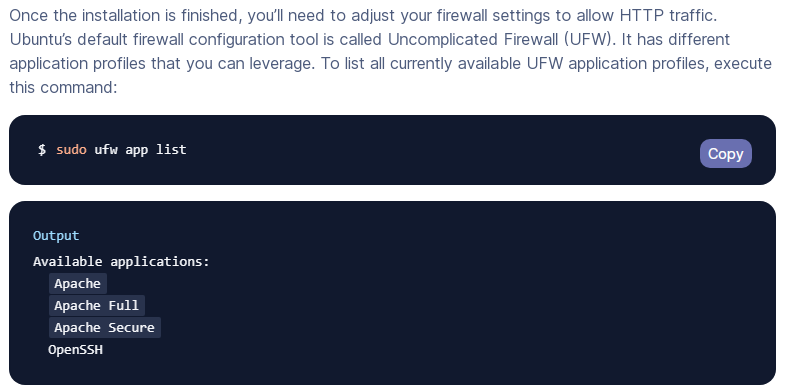
sudo ufw reload

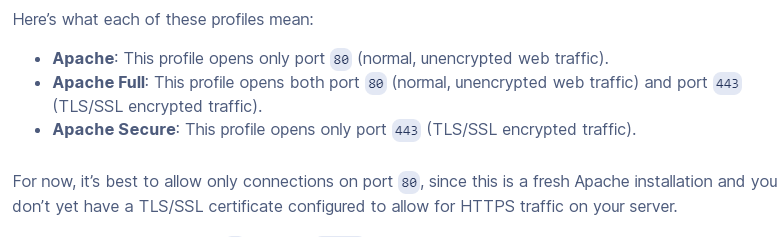
sudo ufw status

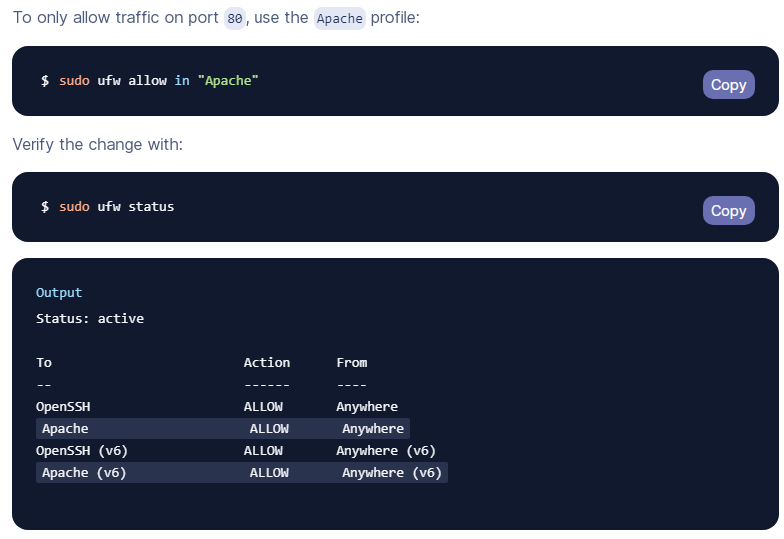
check in browser   [http://server-ip-address](http://server-ip-address/)



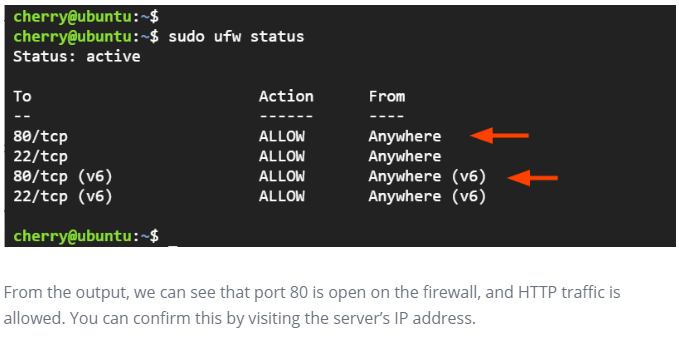
sudo ufw app list

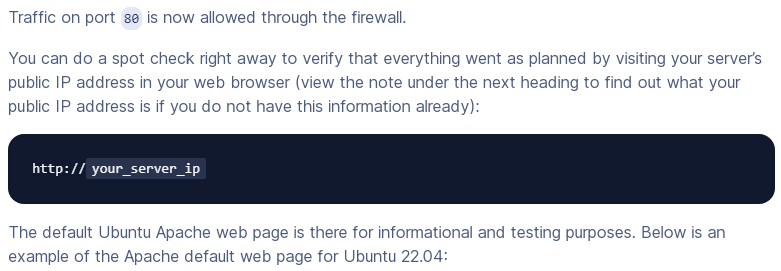




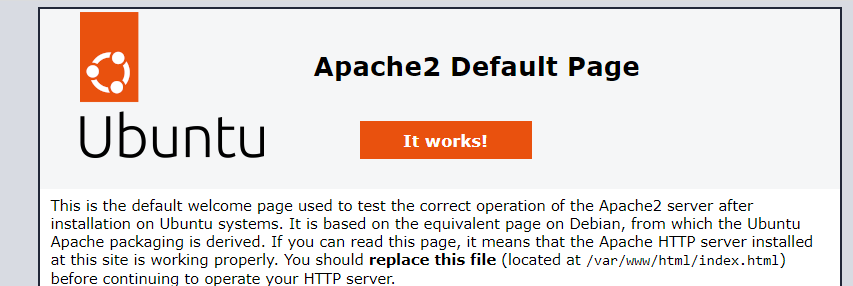


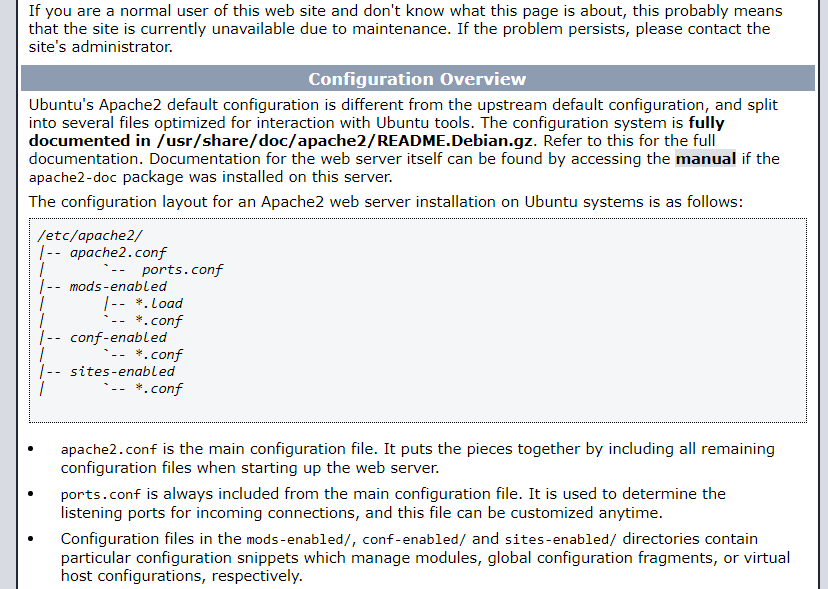


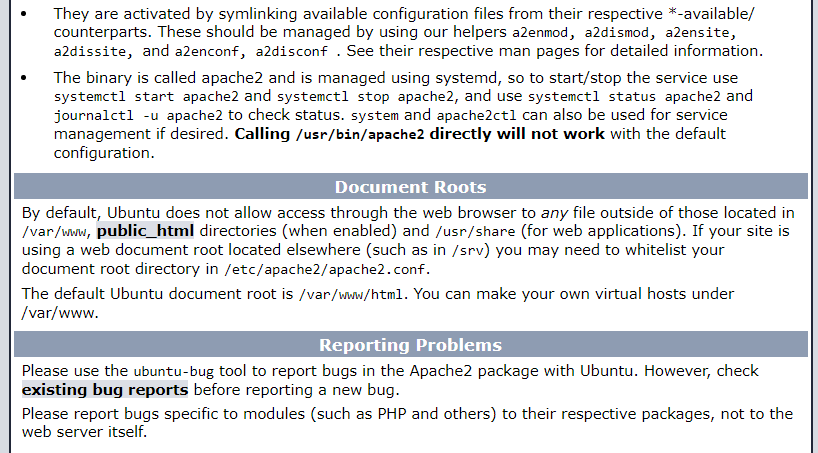










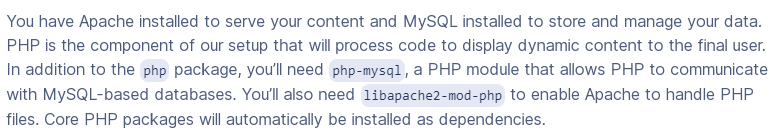


## Step — Installing PHP & PHP MODULES

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

php –v





### Step 4:Install PHP and PHP modules

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

php --version

php -m   //list of module for showing

sudo nano  /var/www/html/info.php

<http://server-ip/info.php>

sudo nano /etc/apache2/mods-enabled/dir.conf

sudo apt install

ghostscript \

libapache2-mod-php \

php \

php-bcmath \

php-curl \

php-imagick \

php-intl \

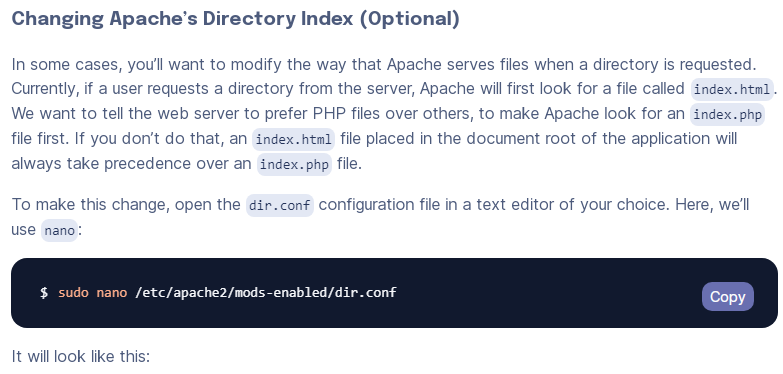
php-json \

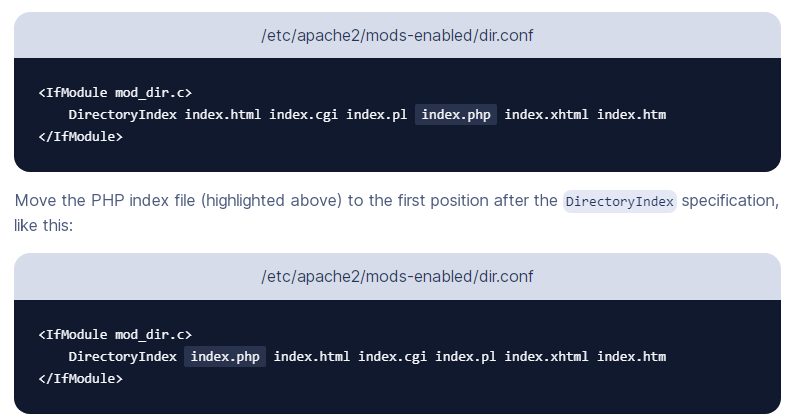
php-mbstring \

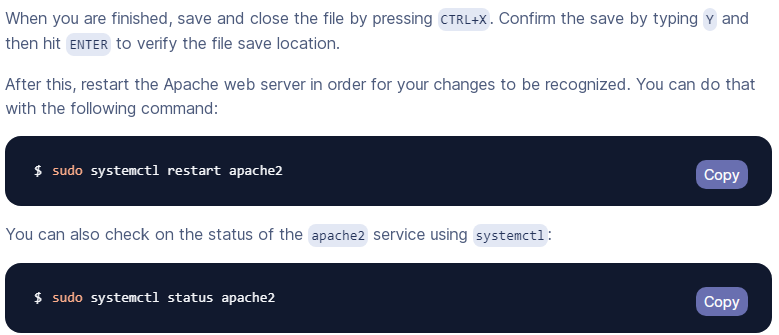
php-mysql \

php-xml \

php-zip







The WordPress package we are going to install is the official release from **WordPress.org**. First we will make the directory and change its permissions to download our installation in.

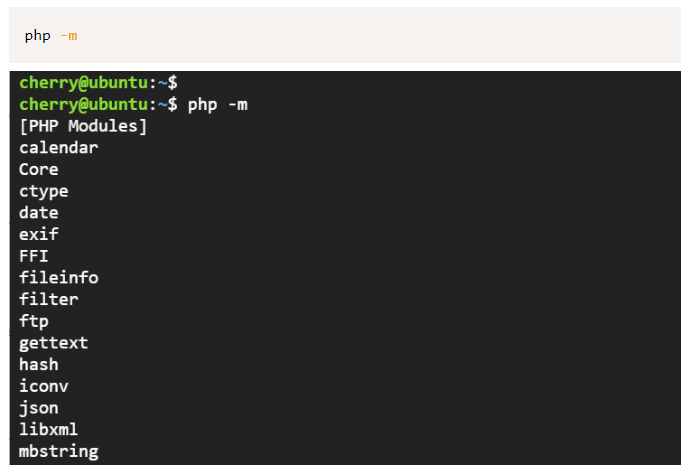
sudo mkdir -p /srv/www

sudo chown www-data: /srv/www



The following instruction downloads the WordPress package.

curl https://wordpress.org/latest.tar.gz | sudo -u www-data tar zx -C /srv/www



## [Step — Installing](https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu#step-3-installing-php) Mariydb

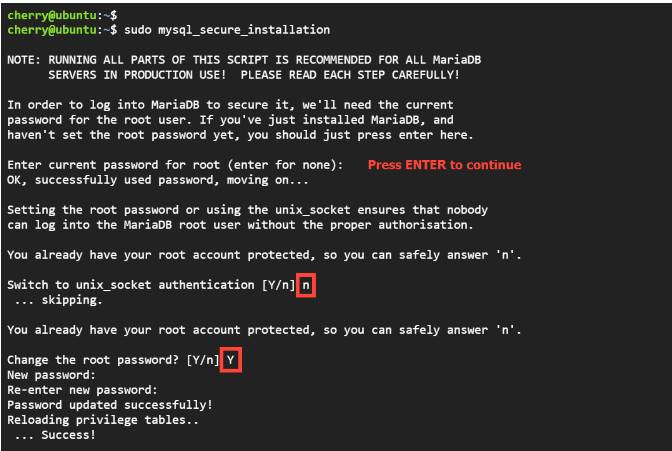
### Step 2: Install MariaDB database server

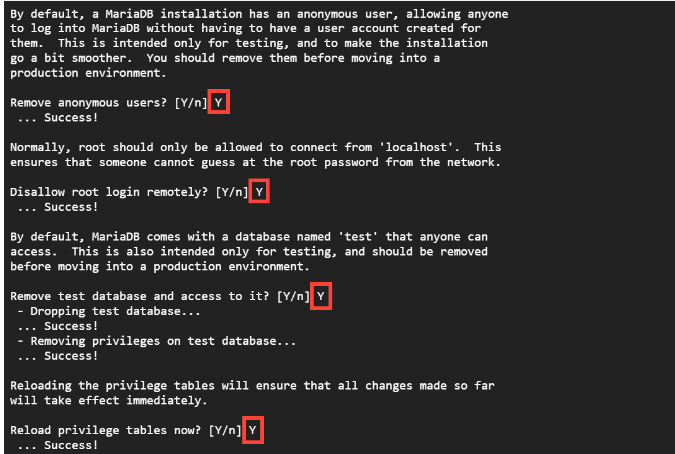
sudo apt install mariadb-server mariadb-client -y

mariadb --version

### Step 2: Secure MariaDB database server

sudo mysql\_secure\_installation





CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';

CREATE DATABASE mydatabase1;

CREATE USER 'rajivuser'@'%' IDENTIFIED BY 'root';

CREATE USER 'rajivuser'@'localhost' IDENTIFIED BY 'root';

For showing user list

SELECT User, Host FROM mysql.user;

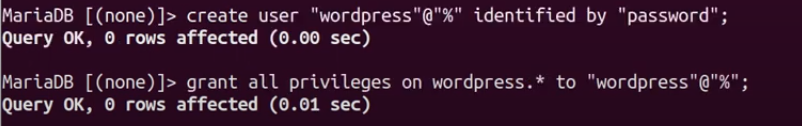
GRANT ALL PRIVILEGES ON \*.\* TO 'rajivuser'@'localhost' WITH GRANT OPTION;

GRANT ALL ON example\_database.\* TO 'example\_user'@'%';

FLUSH PRIVILEGES;

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';



sudo apt install phpmyadmin

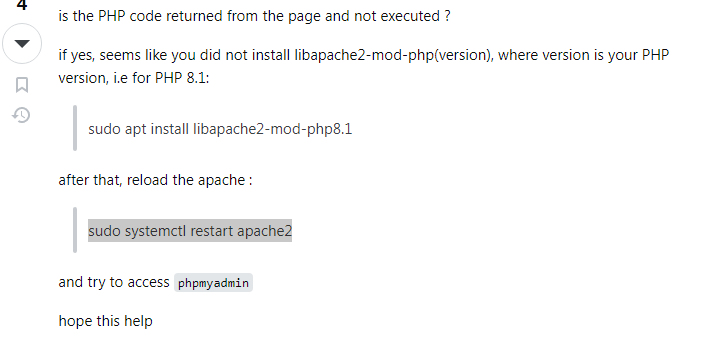
Password Tiger@12345

sudo phpenmod mbstring  
sudo phpenmod mysqli

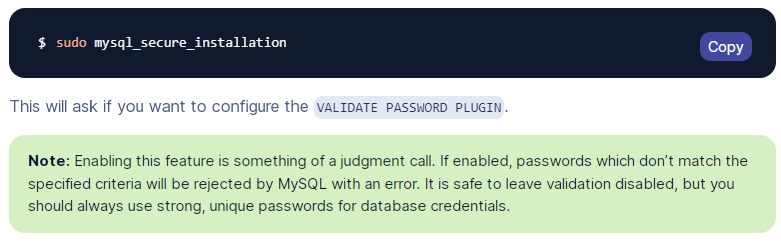
sudo systemctl restart apache2

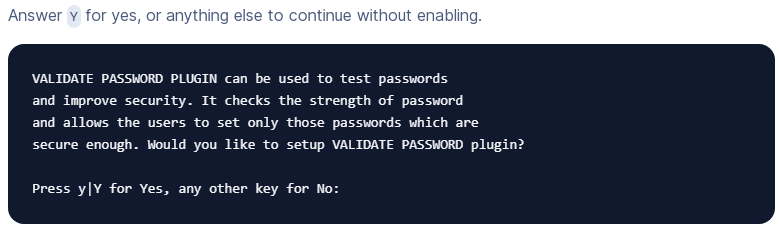
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';

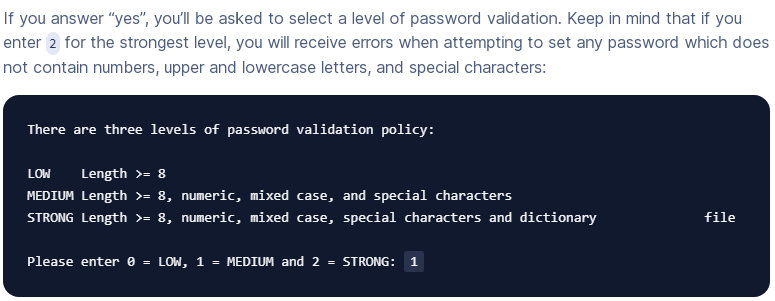
<https://askubuntu.com/questions/1417547/ubuntu-22-04-phpmyadmin-not-working>

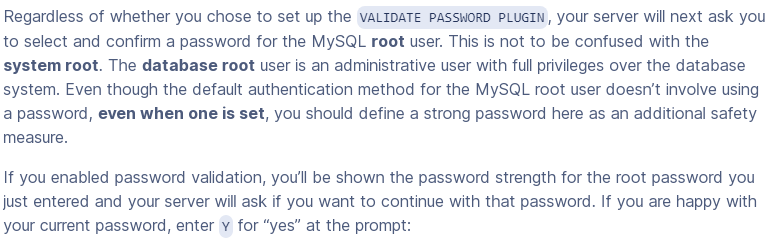


sudo mysql\_secure\_installation











sudo mysql -u root

The above instructions opens MySQL’s own terminal. We create a separate database titled ‘wordpress-db’ and create a user identified by a password of your choosing.

CREATE DATABASE wordpress-db;

CREATE USER wordpress-db@localhost IDENTIFIED BY 'password';

We grant it privileges and flush them.

GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER

FLUSH PRIVILEGES;

exit

To connect the database, we copy the config file and set the configuration in the file. Replace with your database name, db username and password.

sudo -u www-data cp /srv/www/wordpress/wp-config-sample.php /srv/www/wordpress/wp-config.php

sudo -u www-data sed -i 's/database\_name\_here/wordpress/' /srv/www/wordpress/wp-config.php

sudo -u www-data sed -i 's/username\_here/wordpress/' /srv/www/wordpress/wp-config.php

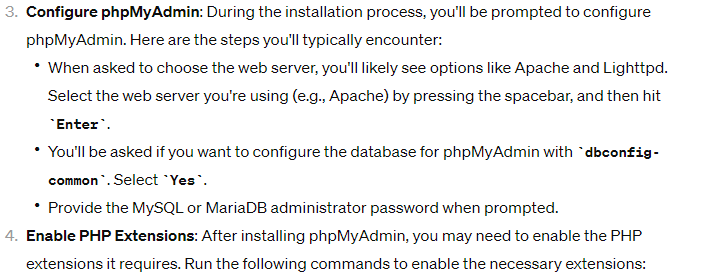
sudo -u www-data sed -i 's/password\_here//' /srv/www/wordpress/wp-config.php

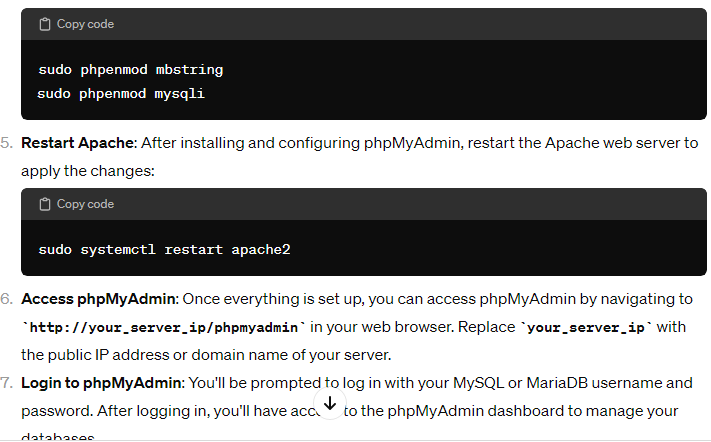
Open the following file using nano.

sudo -u www-data nano /srv/www/wordpress/wp-config.php

### Installation & Configure Of PHPMADMIN

sudo apt install phpmyadmin





### Configure Apache & VITUAL HOST [OPTIONAL]

Now, we need to configure the [Apache web server](https://cloudinfrastructureservices.co.uk/how-to-setup-apache-web-server-mysql-server-on-linux-in-azure-aws-gcp/) site for WordPress. We create a file in the with the path **/etc/apache2/sites-available/wordpress.conf** and copy the following into the conf file (edit your ServerName and ServerAlias).

<VirtualHost \*:80>

ServerAdmin admin@example.com

DocumentRoot /srv/www/wordpress

ServerName sitename.com

ServerAlias www.sitename.com

<Directory /srv/www/wordpress/>

Options FollowSymlinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/wordpress\_error.log

CustomLog ${APACHE\_LOG\_DIR}/wordpress\_access.log combined

</VirtualHost>

**We will now enable the site and enable URL writing.**

sudo a2ensite wordpress

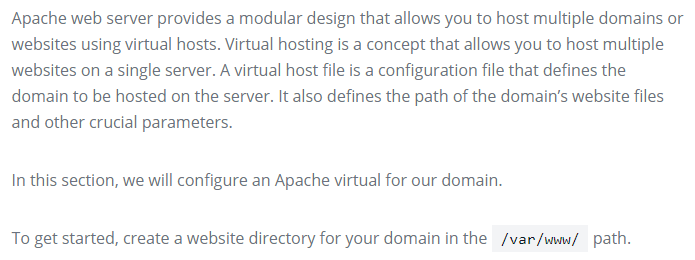
sudo a2enmod rewrite

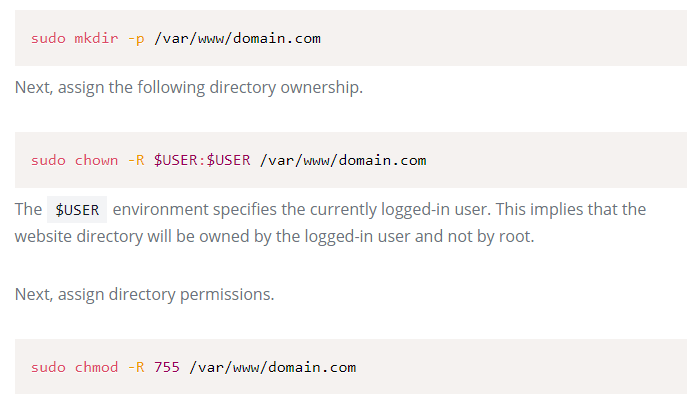
Now, we reload apache2.

sudo service apache2 reload

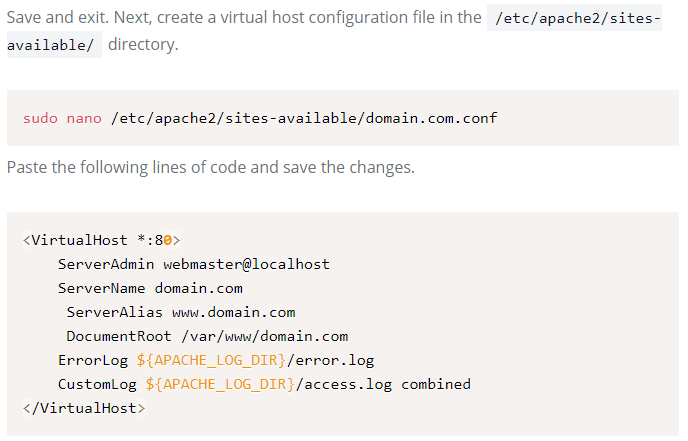
Run the following command to disable the default page so that WordPress loads.

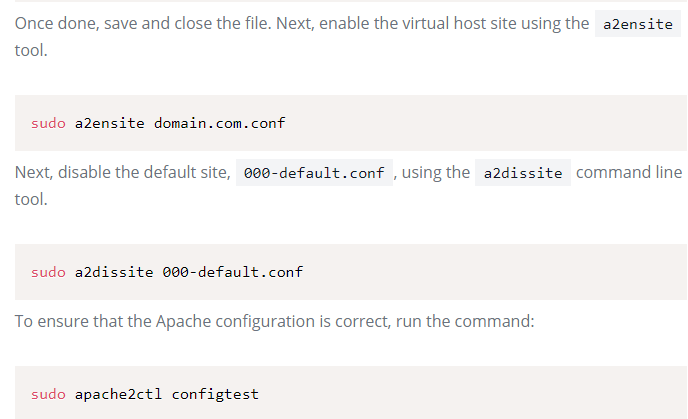
sudo a2dissite 000-default

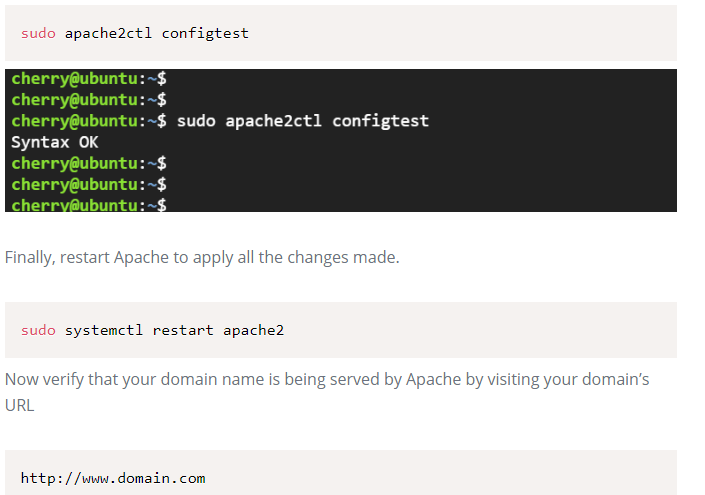




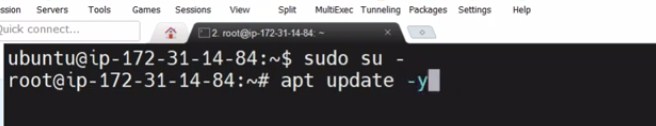


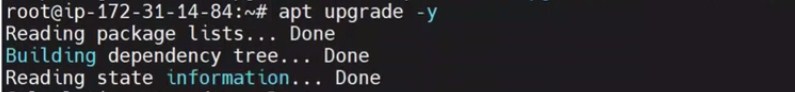


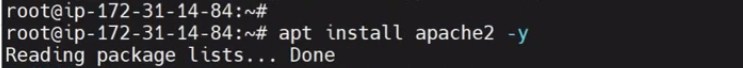


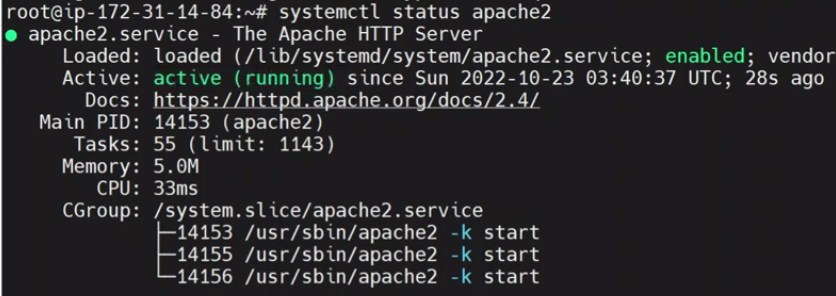


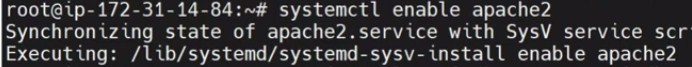
**Wordpress Installation on AWS**

****

****

****

****

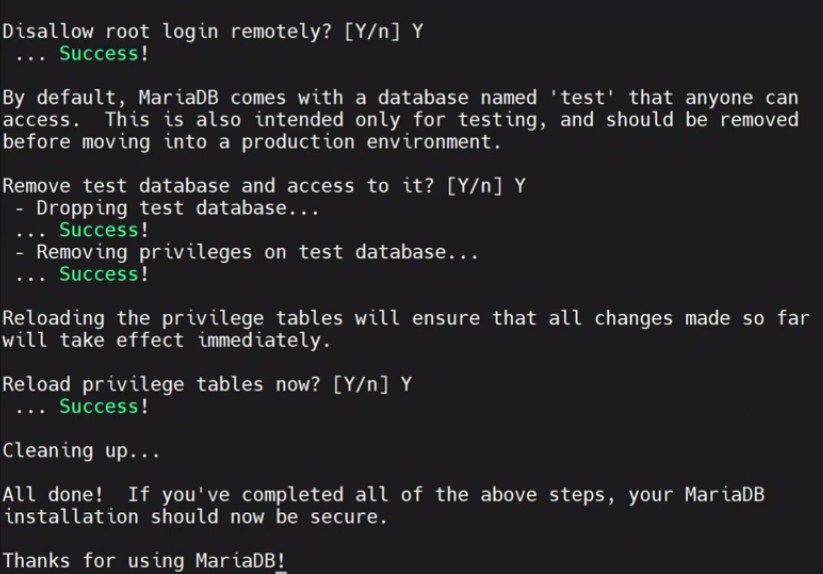
****

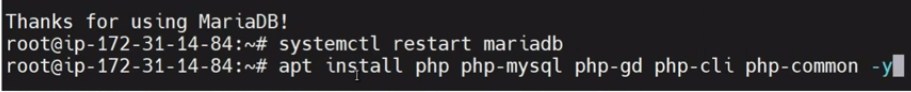
**C:\Users\Rajiv\Desktop\Screenshot_14.jpg**

**C:\Users\Rajiv\Desktop\Screenshot_15.jpg**

**C:\Users\Rajiv\Desktop\Screenshot_16.jpg**

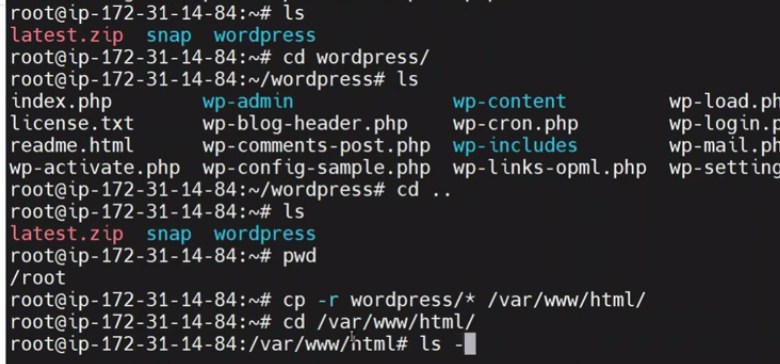
****

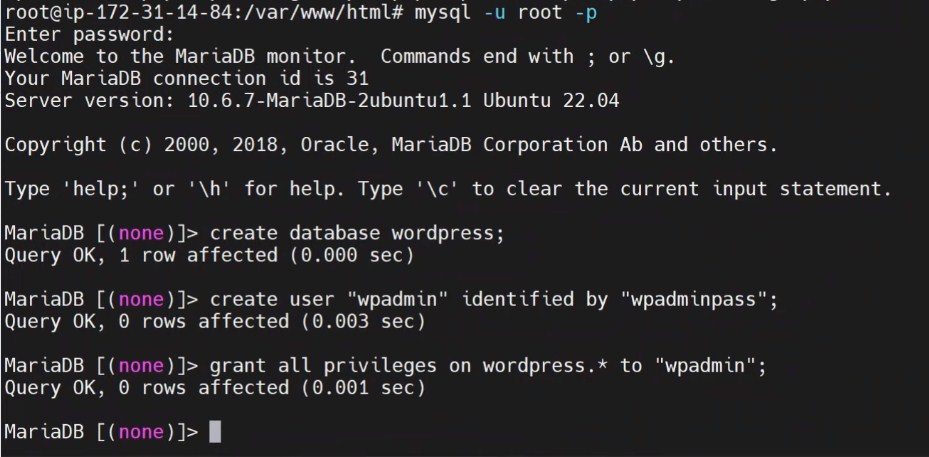
****

****

**C:\Users\Rajiv\Desktop\Screenshot_20.jpg**

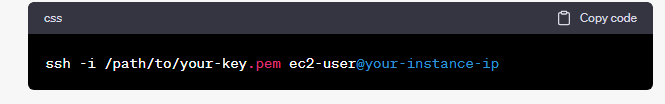
****

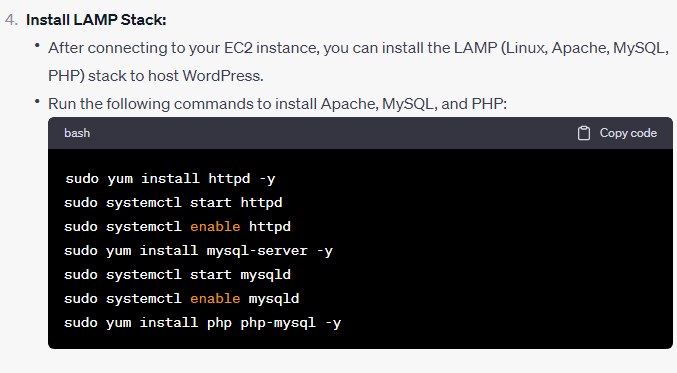
****

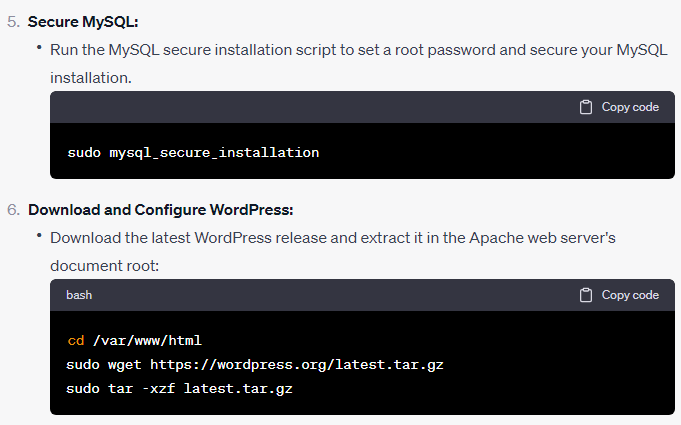
****

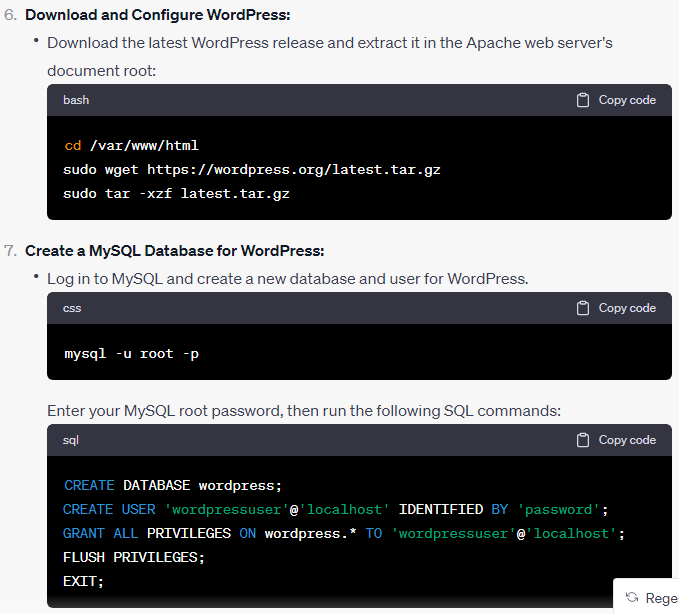
**Wordpres installation on AWS**

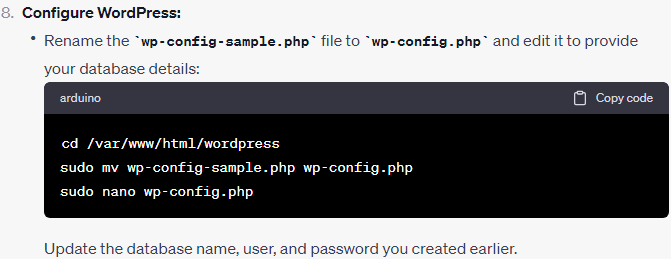
1. **Create an AWS Account:** If you don't already have an AWS account, sign up for one at <https://aws.amazon.com/>. You will need to provide billing information, but you can use the AWS Free Tier to get started without incurring charges for the first 12 months.
2. **Launch an EC2 Instance:**
   * Sign in to your AWS Management Console.
   * Navigate to the EC2 Dashboard.
   * Click the "Launch Instance" button to create a new virtual server.
   * Choose an Amazon Machine Image (AMI) based on your requirements. You can use an Amazon Linux AMI, which is suitable for most purposes.
   * Select an instance type, which determines the server's hardware resources. The t2.micro instance is part of the Free Tier.
   * Configure instance details, such as the number of instances, network settings, and storage.
   * Add storage for your instance. You can use the default settings or adjust the storage size as needed.
   * Configure security groups to control incoming traffic to your instance. At a minimum, allow SSH (port 22) and HTTP/HTTPS (ports 80 and 443).
   * Review and launch the instance, and create a new key pair or use an existing one to securely connect to your instance.
3. **Connect to Your EC2 Instance:**
   * Once the instance is running, use SSH to connect to it. Use the private key from the key pair you selected during instance launch.
   * The command to connect will look like this:

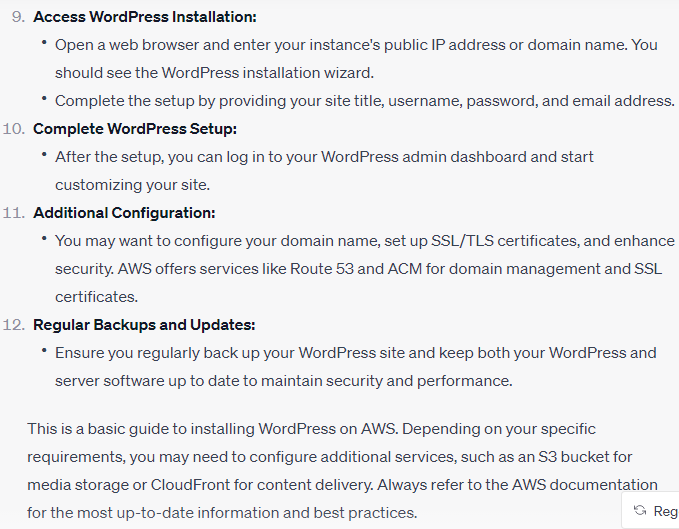
****

****

****

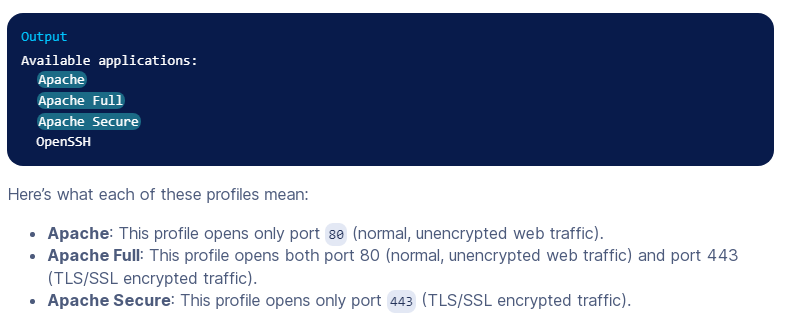
****

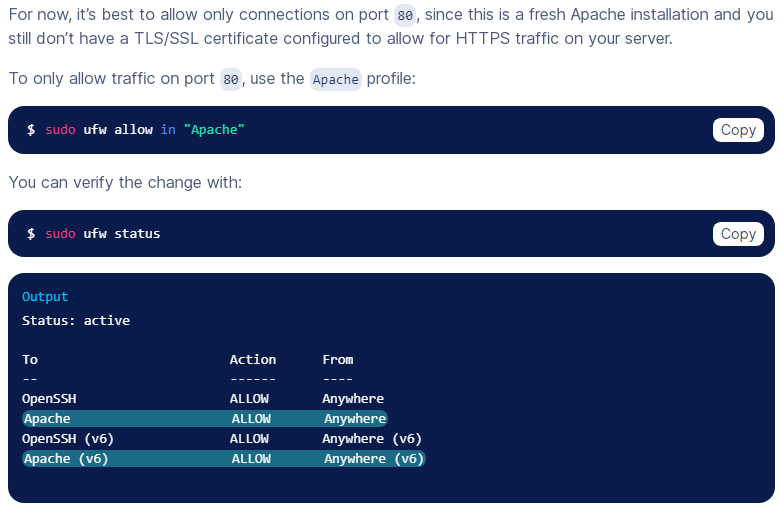
****

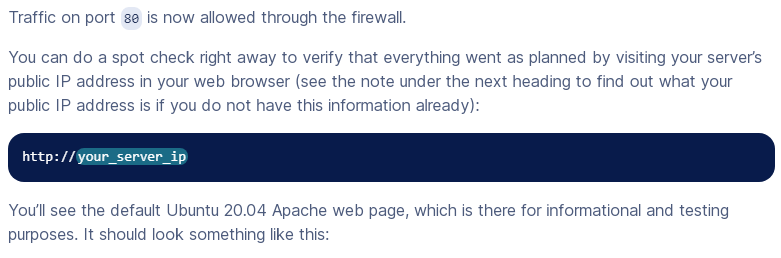
****

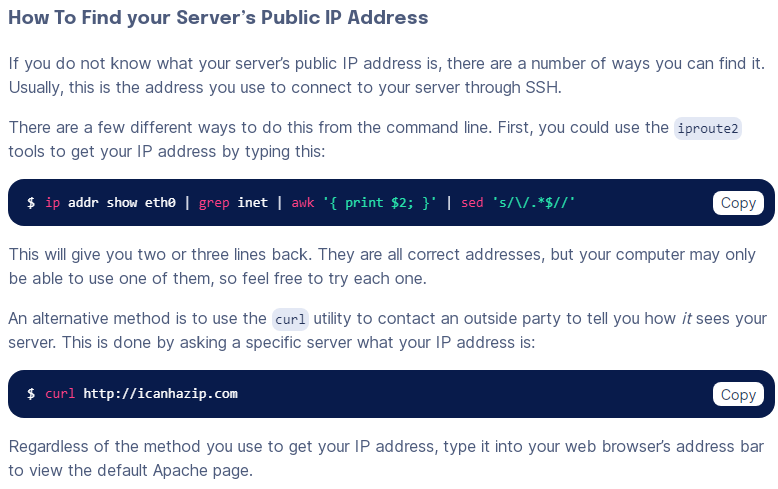
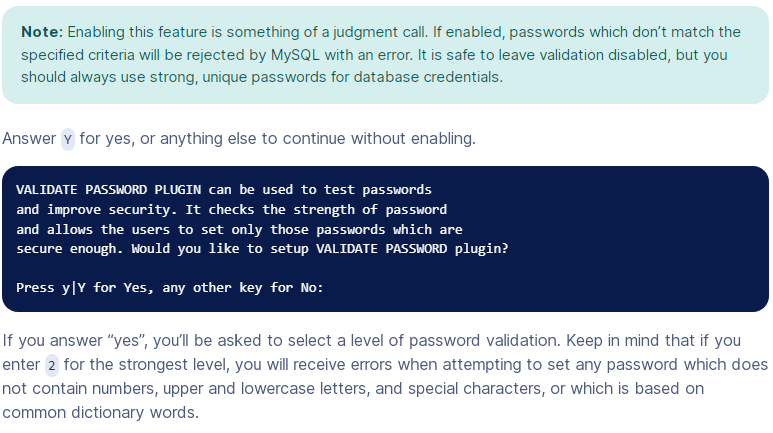
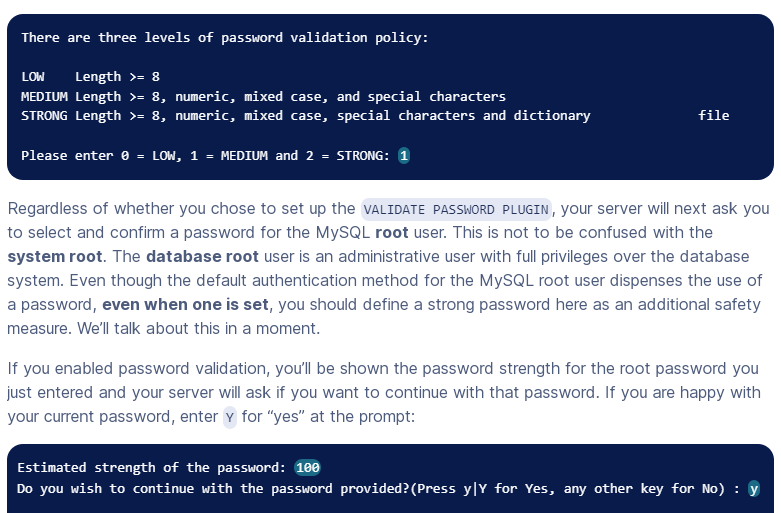
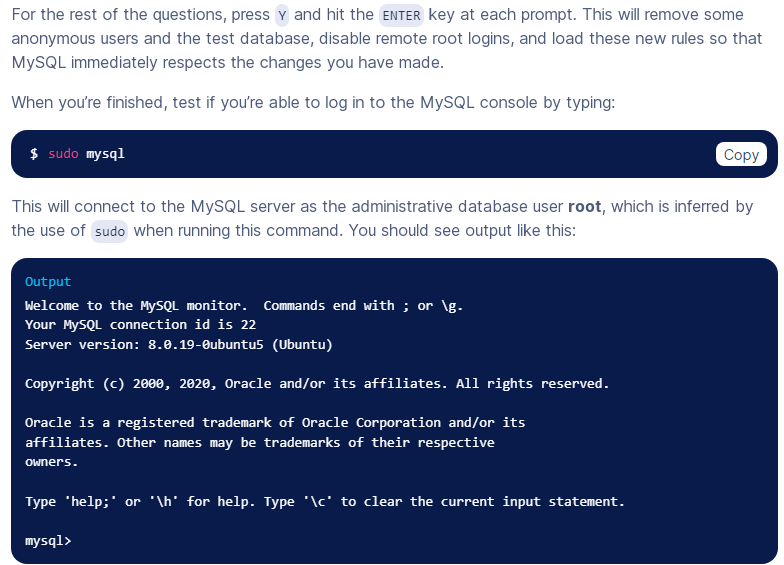
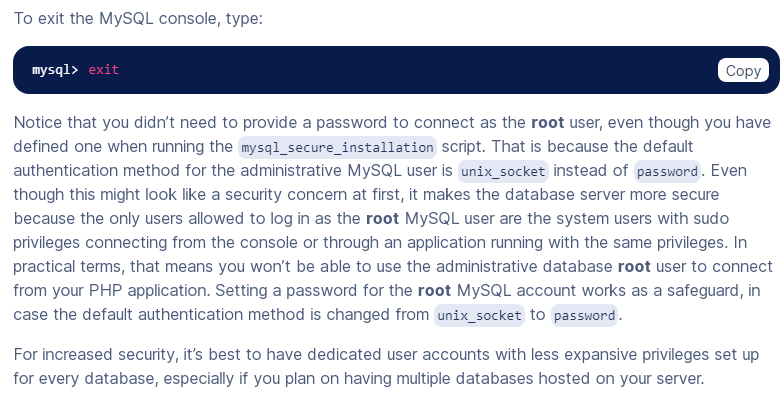
**LAMP Setup**

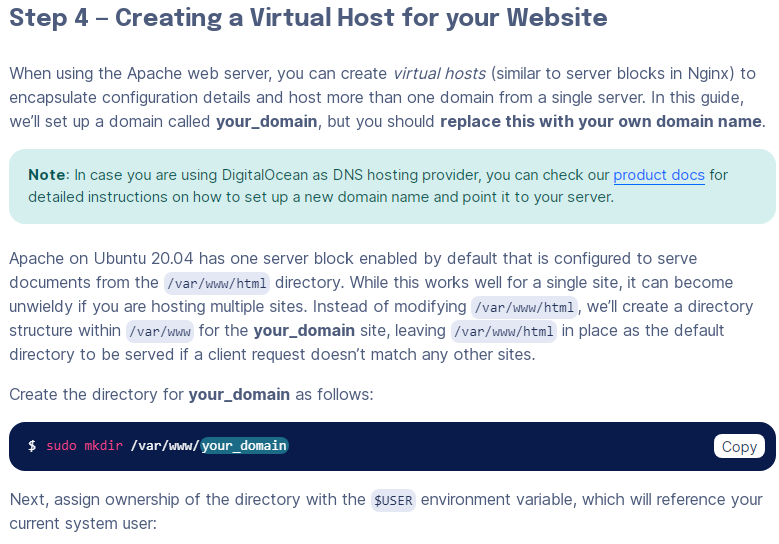
<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04>

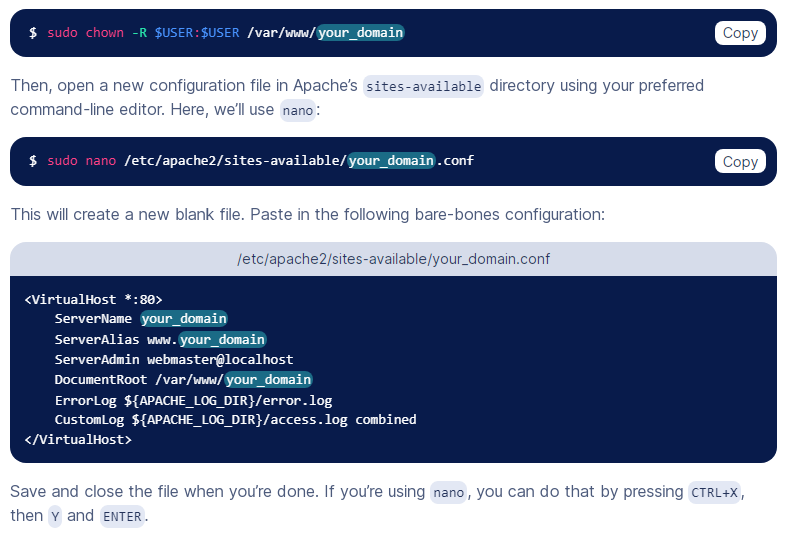
****

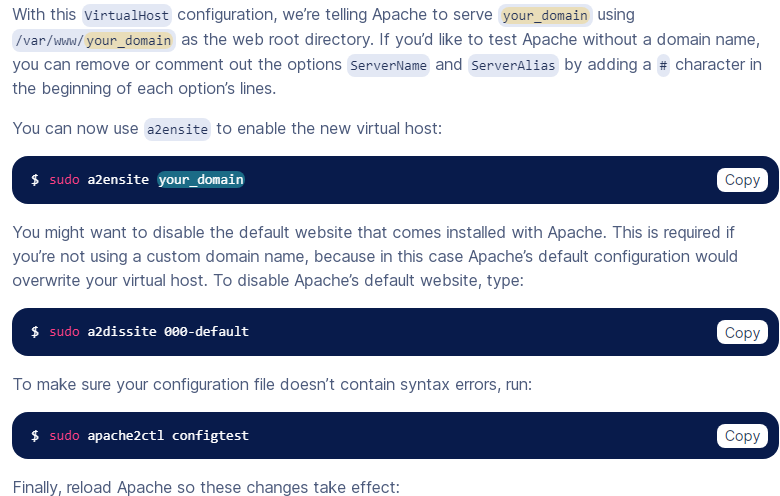
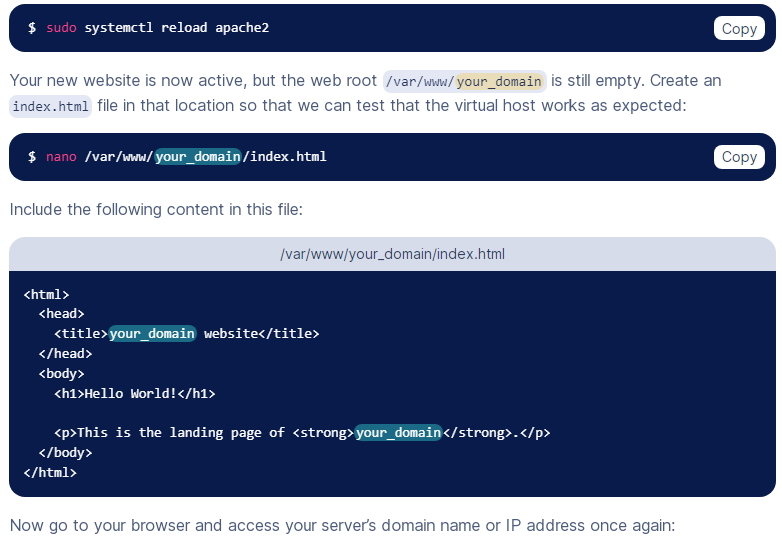
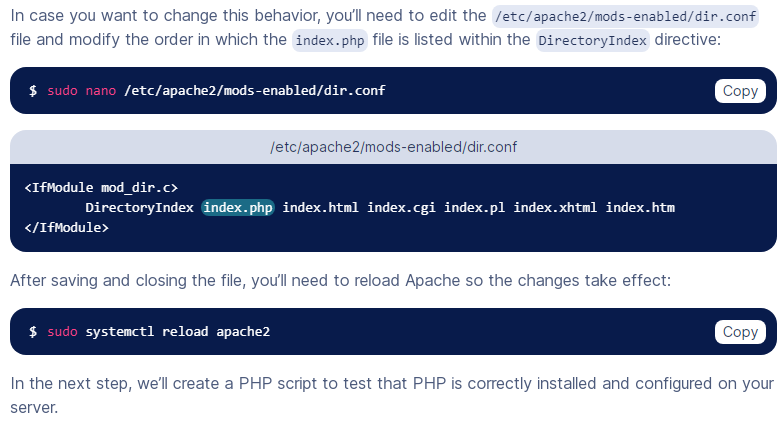
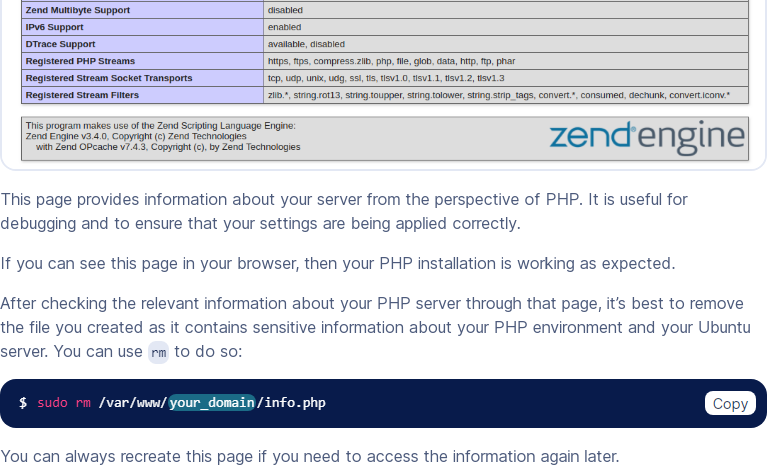
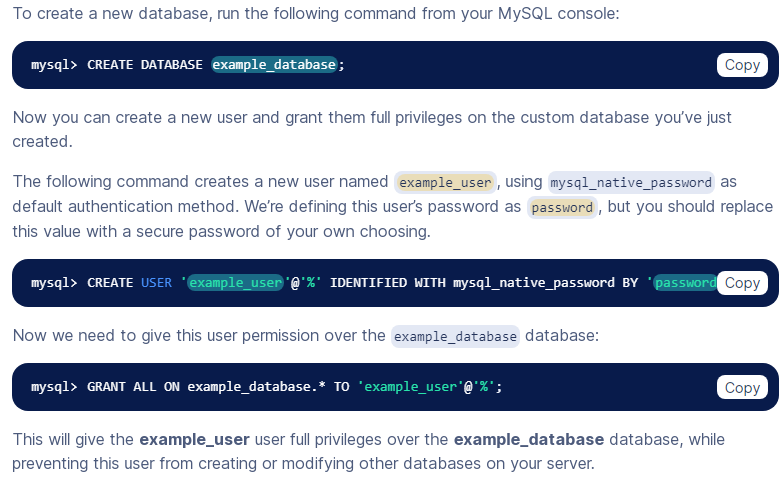
****

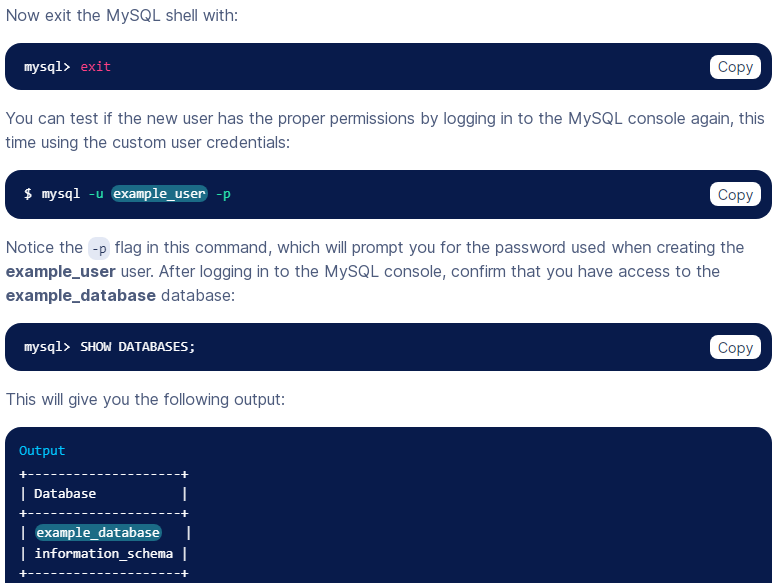
****

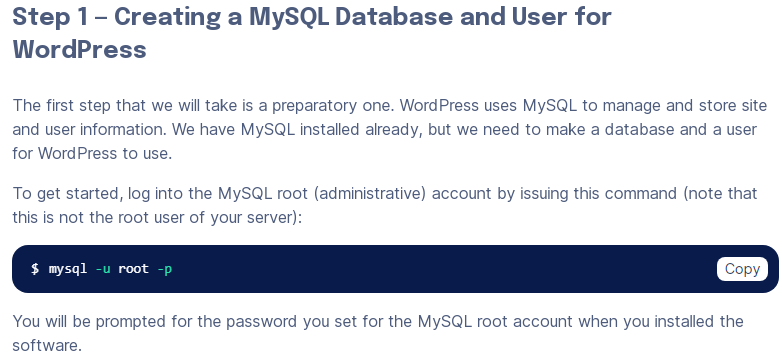
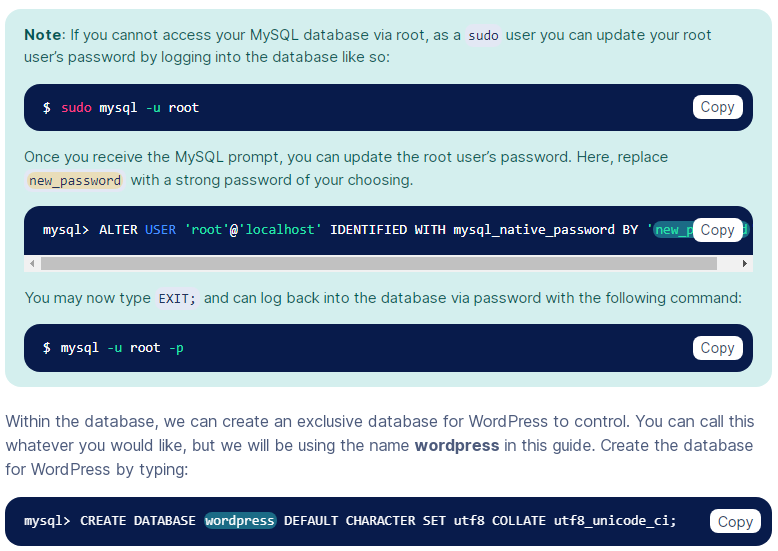
**      **

****

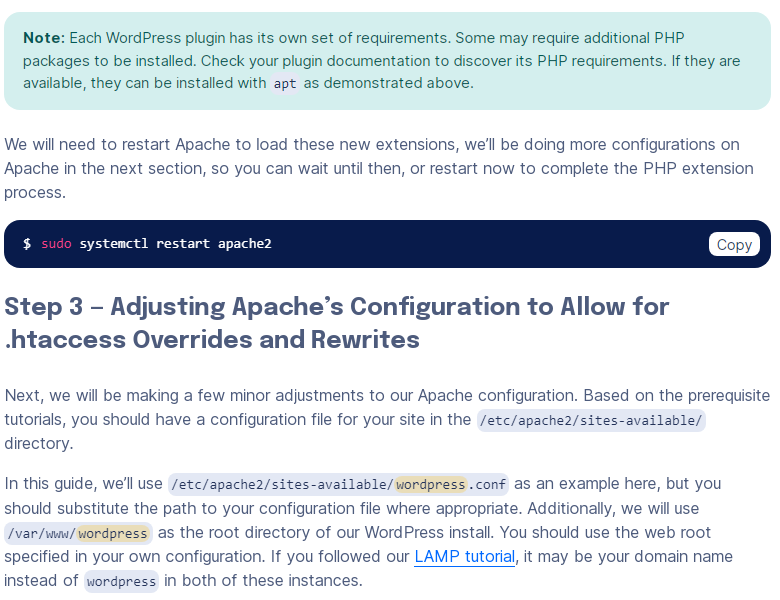
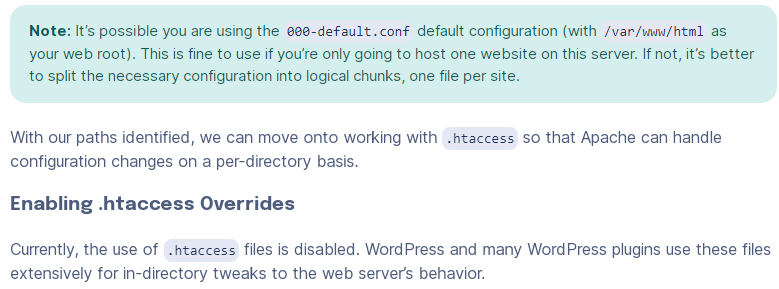
****

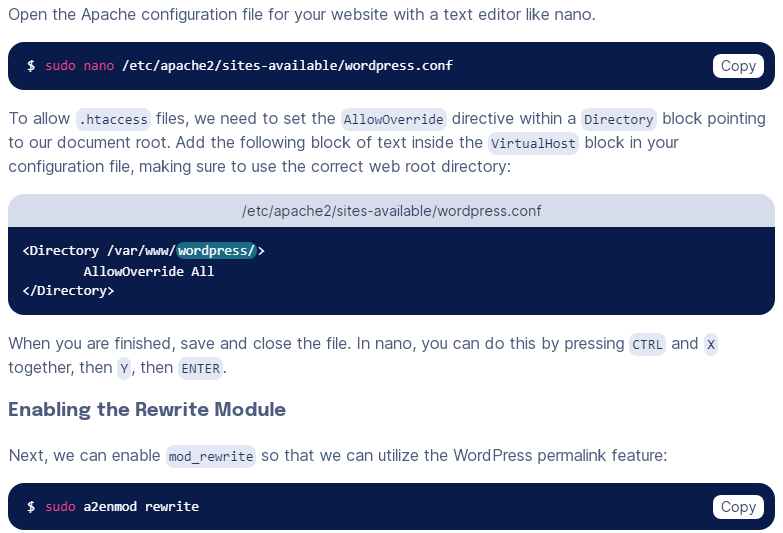
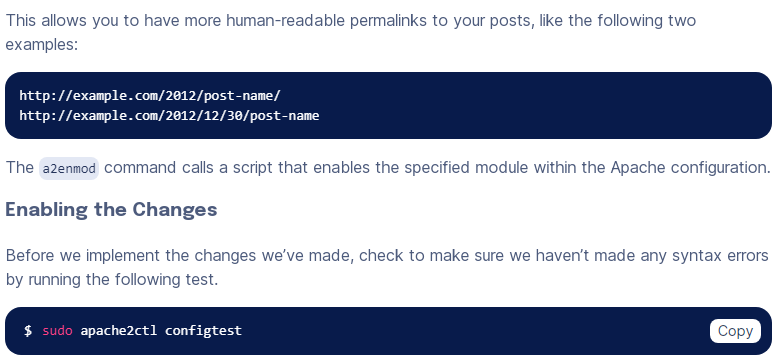
**    **

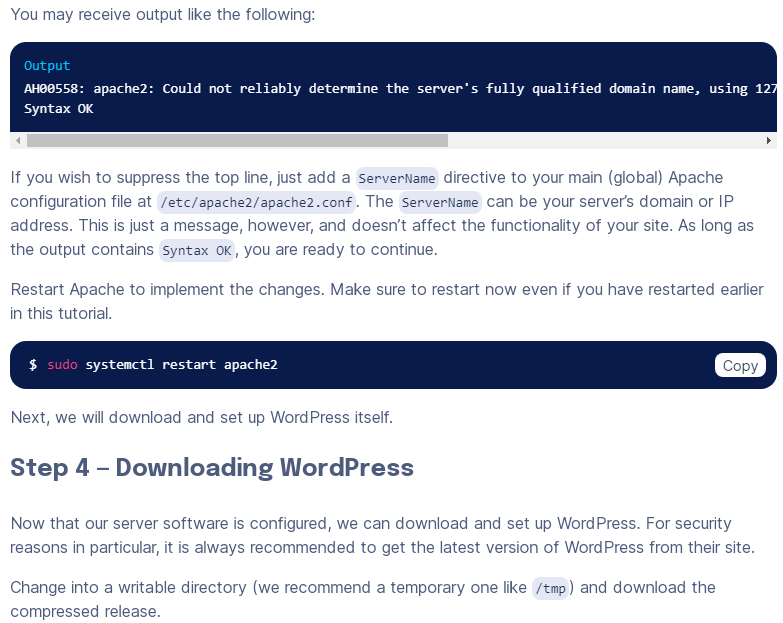
****

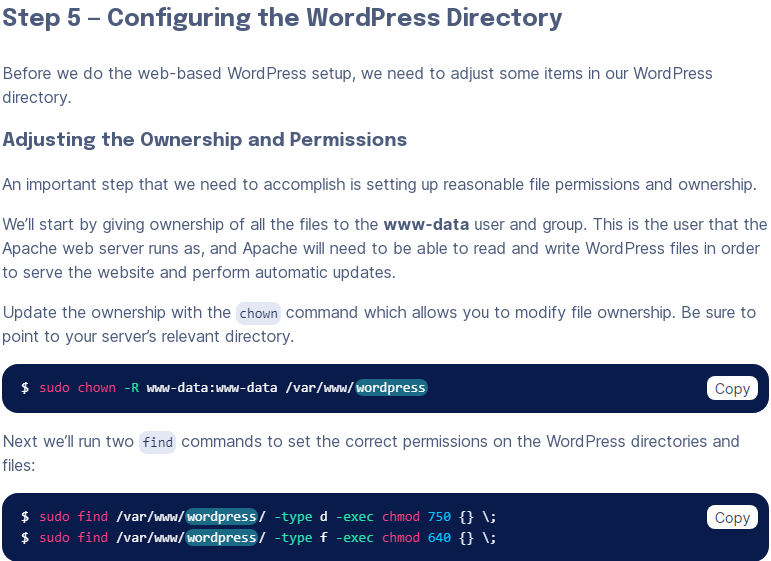
** **

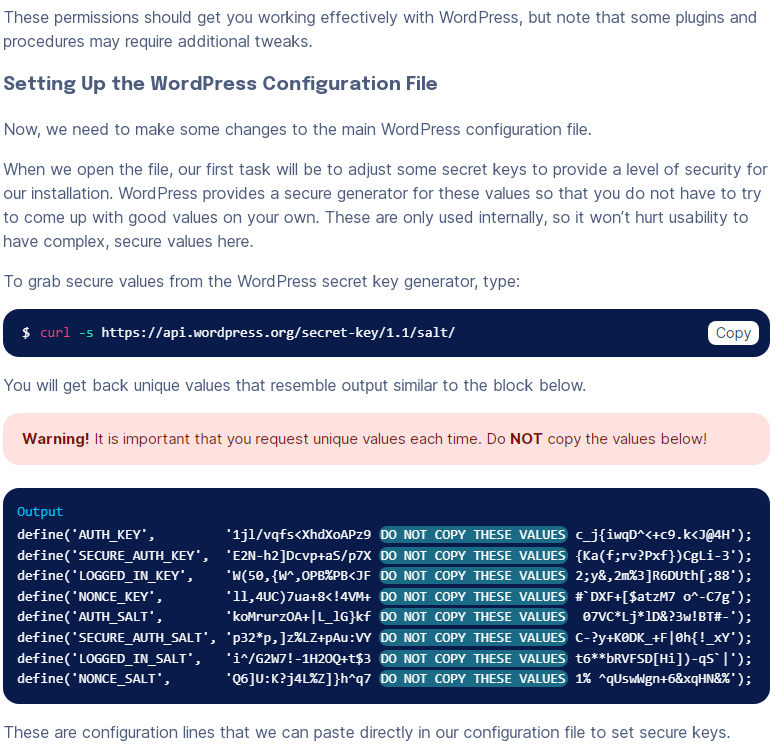
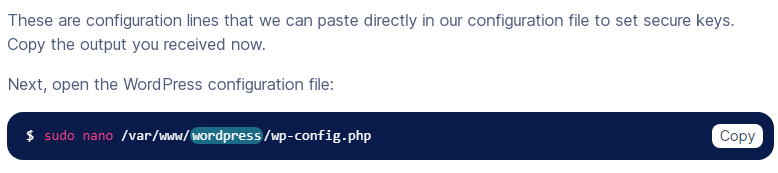
****

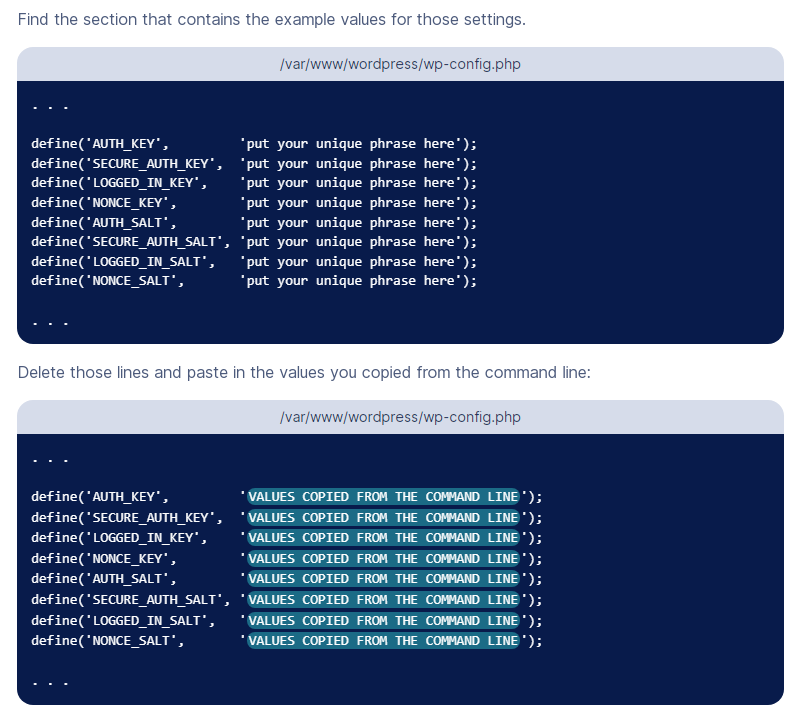
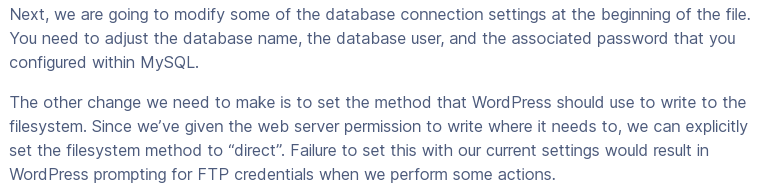
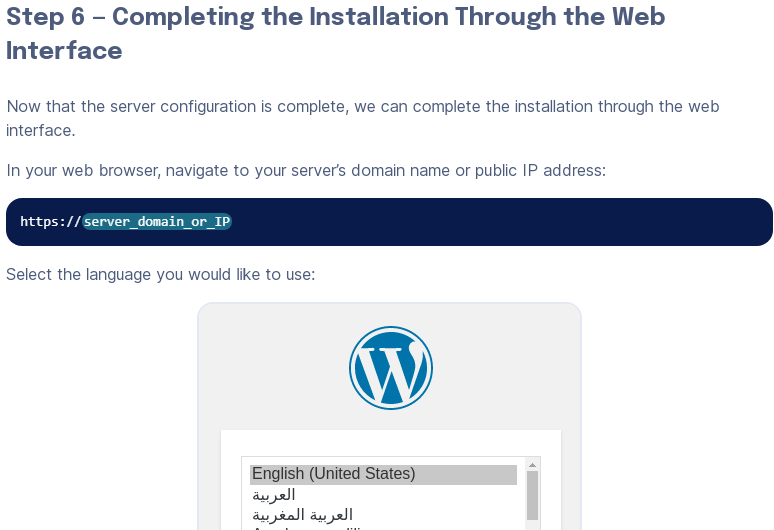
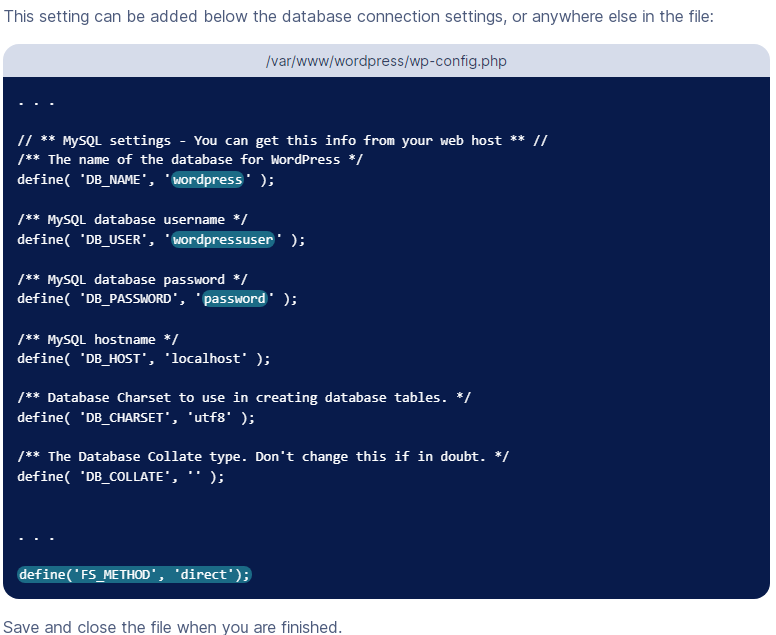
** **

** **

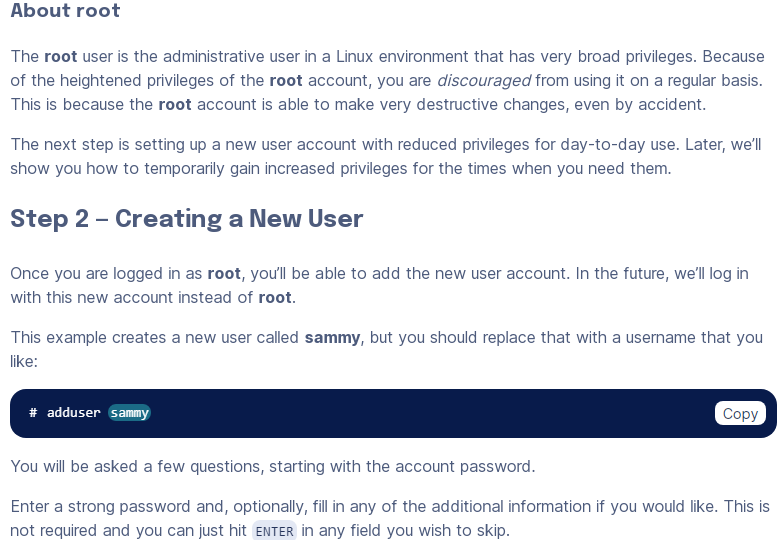
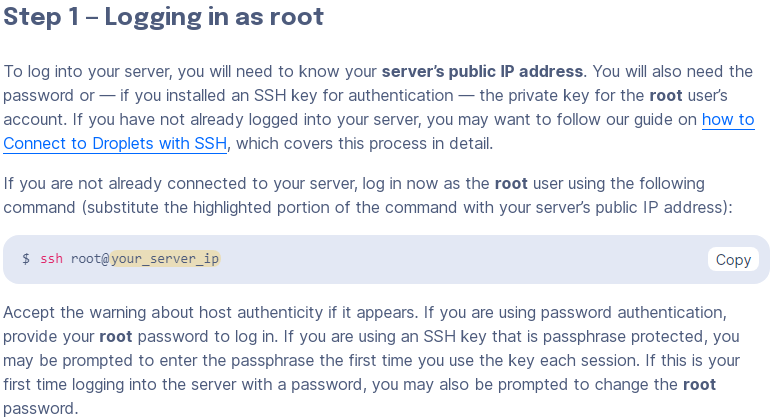
** **

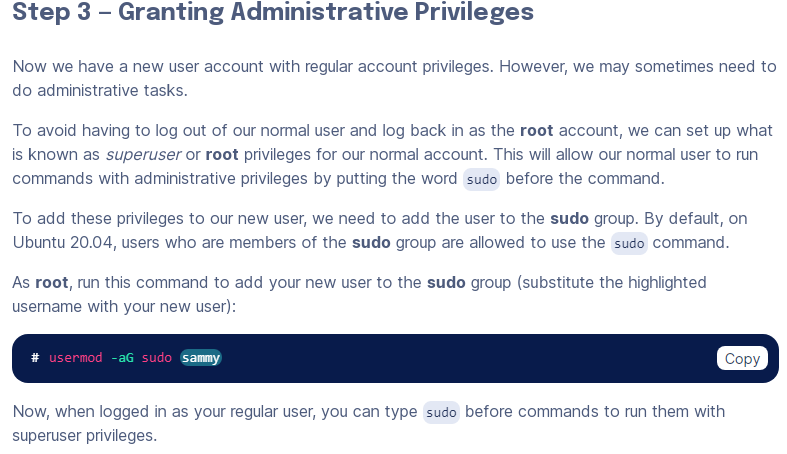
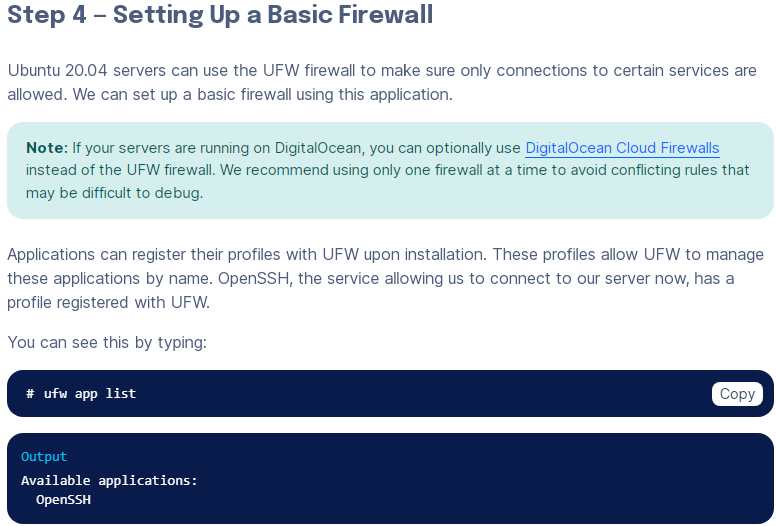
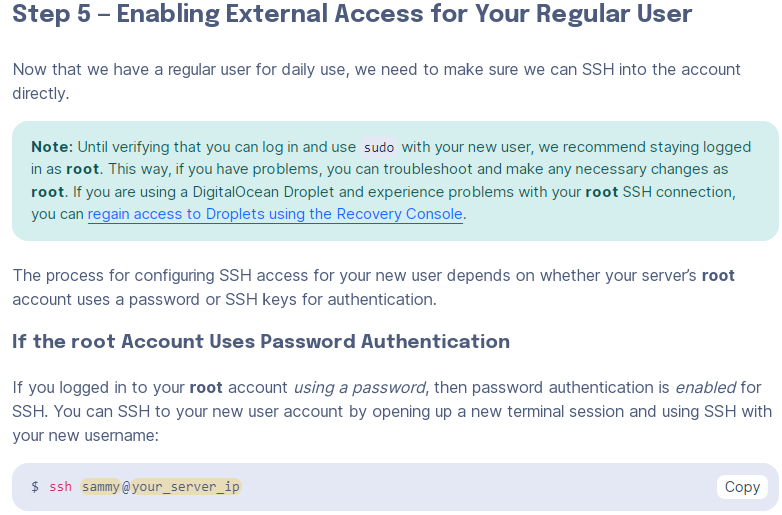
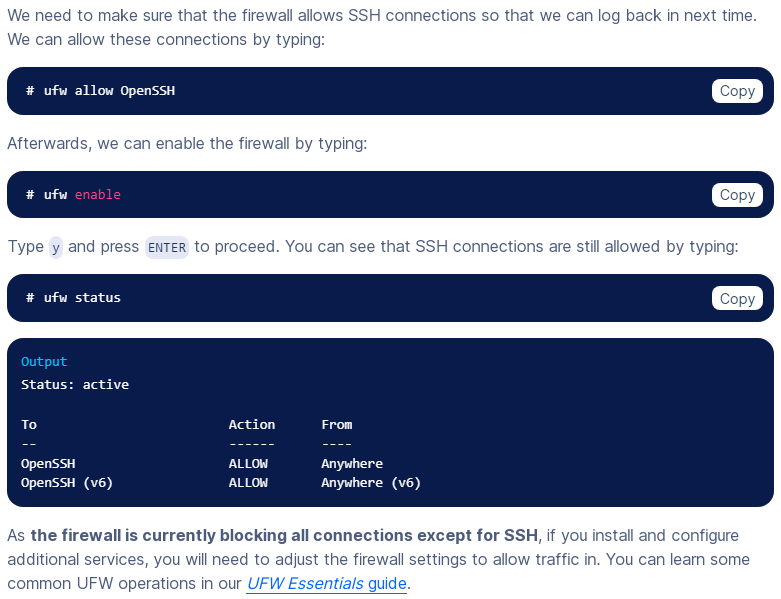
** **

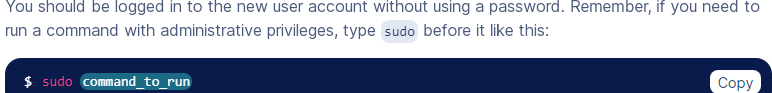
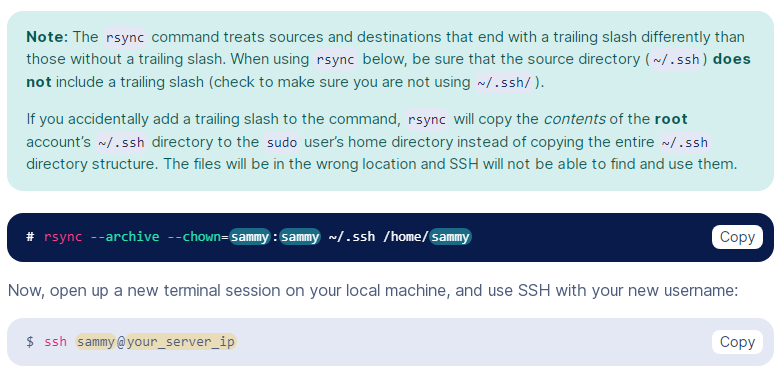
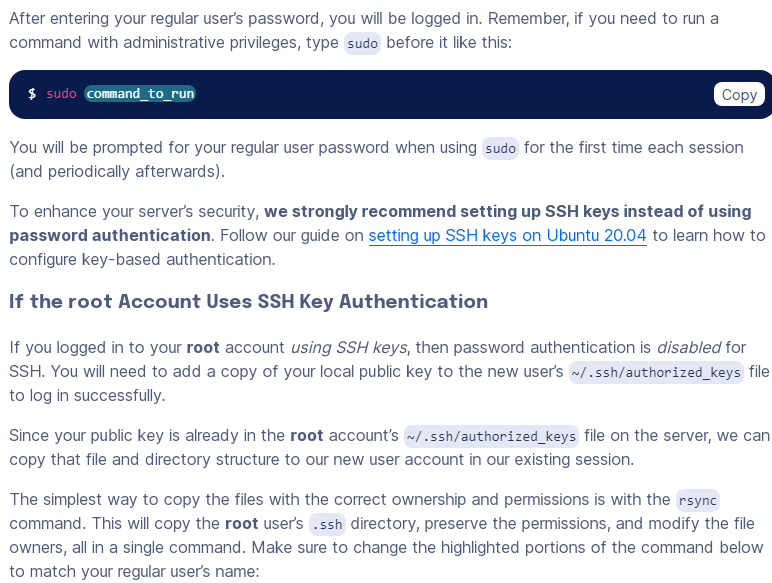
** **

**  **

# Initial Server Setup with Ubuntu 20.04

****

**  **

****

**Install wordpress on windaw iis server**

**Step 1: Download WordPress**

1. Go to the official WordPress website (<https://wordpress.org/download/>) and download the latest version of WordPress in ZIP format.

**Step 2: Extract the WordPress Files**

1. Extract the contents of the ZIP file you downloaded to a directory on your Windows server where you want to host your WordPress site. This will be your website's root directory (e.g., **C:\inetpub\wwwroot\mywordpress**).

**Step 3: Create a MySQL Database**

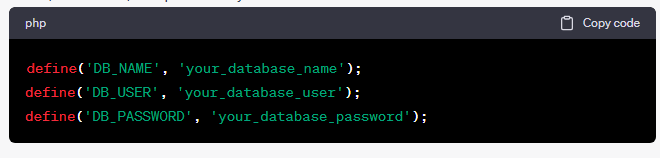
1. You'll need a MySQL database to store WordPress data. Install MySQL on your Windows server if you haven't already.
2. Create a new MySQL database for WordPress. You can use a MySQL client like phpMyAdmin or the MySQL command line to create the database.
3. Create a MySQL user and grant it privileges on the WordPress database. Note down the database name, username, and password for later use.

**Step 4: Configure PHP on IIS**

1. Install PHP on your Windows server. You can use a package like XAMPP or install PHP manually.
2. Configure PHP to work with IIS by adding PHP as a FastCGI module. You can use the "Internet Information Services (IIS) Manager" to configure this.

**Step 5: Configure WordPress**

1. Rename the **wp-config-sample.php** file in your WordPress directory to **wp-config.php**.
2. Open **wp-config.php** and configure the database settings by entering the database name, username, and password you created earlier:



1. You may also want to define other constants like the authentication keys and salts. You can generate these keys by visiting the WordPress.org secret key generator (<https://api.wordpress.org/secret-key/1.1/salt/>).
2. Save your **wp-config.php** file.

**Step 6: Configure IIS for WordPress**

1. Open the "Internet Information Services (IIS) Manager."
2. In the Connections pane on the left, expand your server node and click on "Sites."
3. Right-click on "Default Web Site" or create a new site if needed, and select "Add Application."
4. Set an alias for your WordPress site (e.g., "mywordpress") and browse for the physical path where you extracted the WordPress files.
5. In the "Edit Application" dialog, go to the "Modules" section, and ensure that the "FastCgiModule" is installed and enabled.

**Step 7: Access Your WordPress Site**

1. Open a web browser and enter the URL to your WordPress site (e.g., **http://localhost/mywordpress**). Follow the WordPress installation instructions.
2. Complete the installation by providing your site title, username, password, and email address.
3. Once the installation is complete, you can log in to the WordPress admin dashboard and start customizing your site.

That's it! You've successfully installed WordPress on your Windows IIS server. You can now start building and managing your website using the WordPress platform.