Install lamp on ubuntu 18.04

Step 1 — Installing Apache

Install Apache using Ubuntu’s package manager, apt:

sudo apt update

sudo apt install apache2

step2 – check the status of apache server

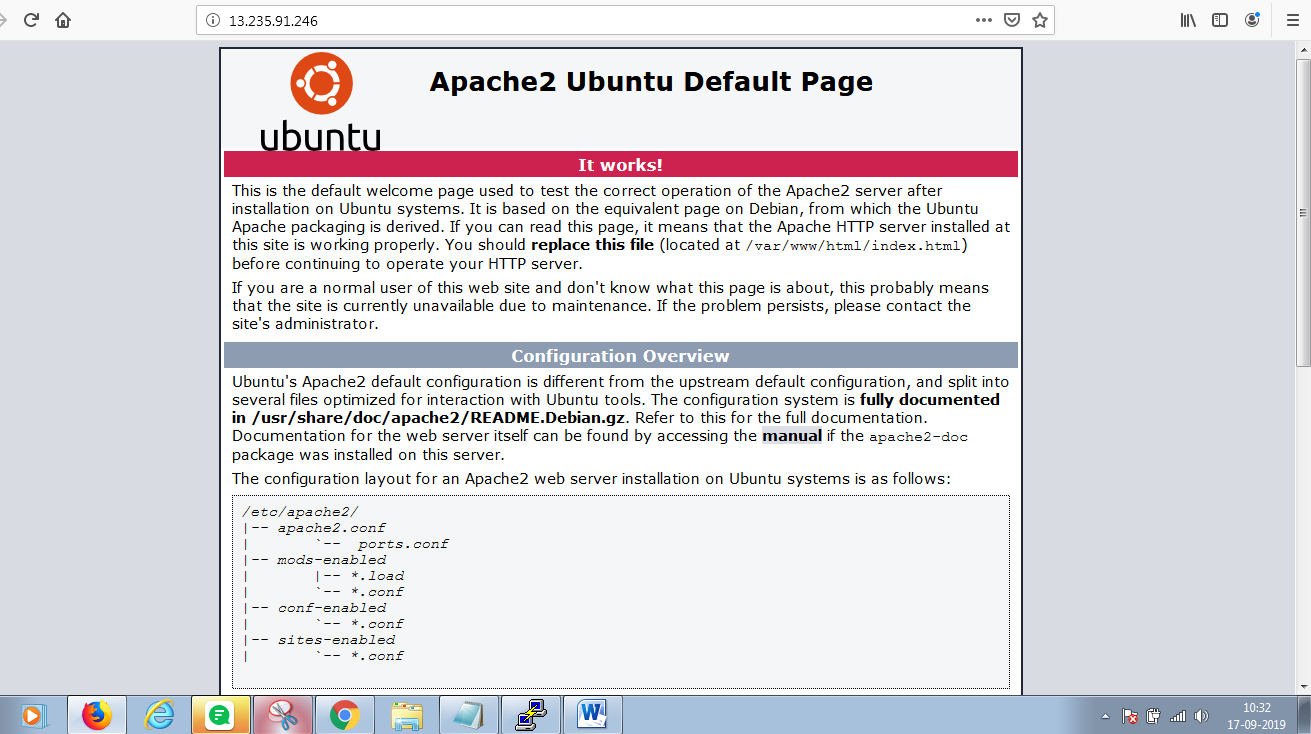
sudo service apache2 status

step3– Restart the apache server

sudo service apache2 restart

sudo service apache2 stop

sudo service apache2 start



### Install MySQL and Create a Database

Step1:To install MySQL, type in the following command:

sudo apt-get install mysql-server

step2: Run the mysql\_secure\_installation script to address several security concerns in a default MySQL installation.

sudo mysql\_secure\_installation

Step3: To log in to MySQL as the root user:

mysql -u root –p

Enter the passowrd and type Exit to come out from mysql.

### Step 4: Install PHP

https://phoenixnap.com/kb/how-to-install-lamp-stack-on-ubuntu

1. To install PHP, run the following command:

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

2. Next, you should modify the way Apache serves files when directories are requested. By default, Apache looks first for a file card named **index.html**. However, we want it to prefer the **index.php** file.

To change this, open the dir.conf file in a text editor with root privileges:

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

step3:You now need to move the PHP index file to the first position:



Step 4: In order for the changes to take effect, you must restart the Apache service.

sudo systemctl restart apache2

step 5: To create a file in that directory, type in the following command in the terminal:

sudo nano /var/www/html/info.php

step 6: Inside the file, type in the valid PHP code:

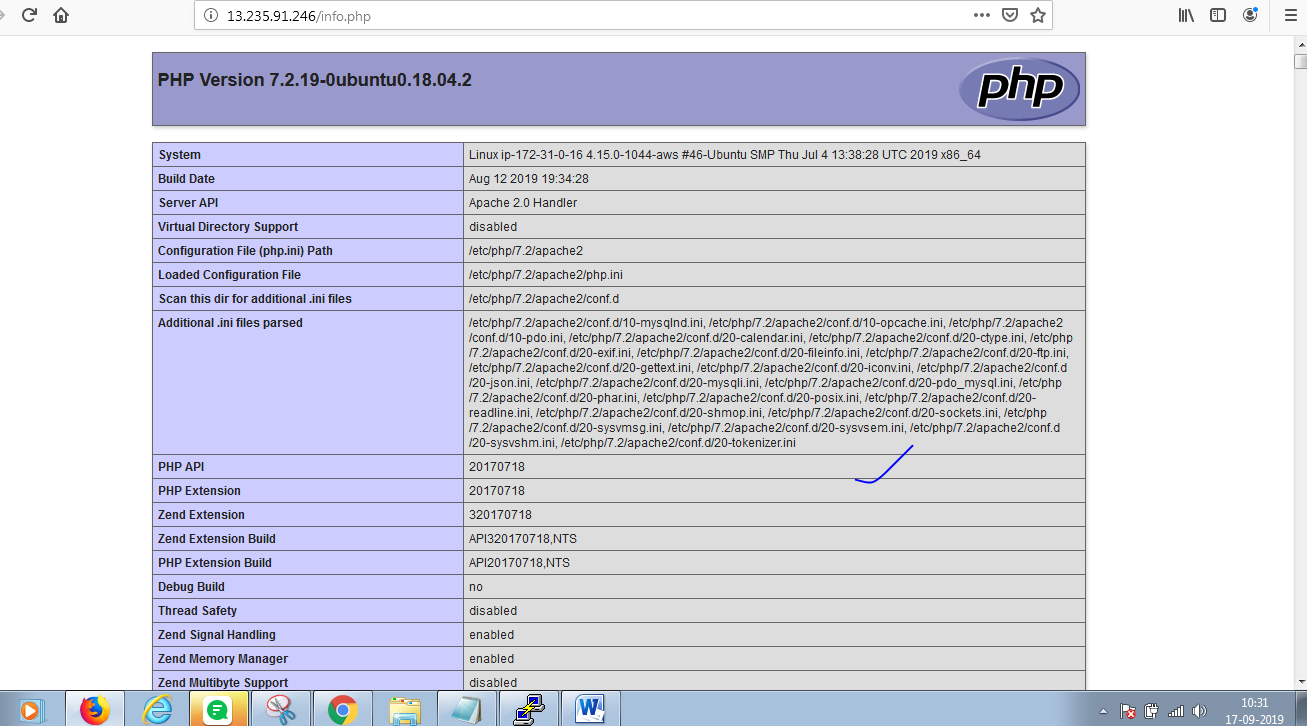
<?php

phpinfo ();

?>

Press **CTRL + X** to save and close the file, and press **y** and **ENTER** to confirm.

Step7: Open a browser and type in your **IP address/info.php**



# Install and Secure phpMyAdmin on Ubuntu 18.04

Link: https://www.digitalocean.com/community/tutorials/how-to-install-and-secure-phpmyadmin-on-ubuntu-18-04

**Step 1 — Installing phpMyAdmin** To get started, we will install phpMyAdmin from the default Ubuntu repositories.

sudo apt update

sudo apt install phpmyadmin php-mbstring php-gettext

This will ask you a few questions in order to configure your installation correctly.

## 

 For the server selection, choose apache2

 Select Yes when asked whether to use dbconfig-common to set up the database

 You will then be asked to choose and confirm a MySQL application password for phpMyAdmin

The installation process adds the phpMyAdmin Apache configuration file into the /etc/apache2/conf-enabled/ directory, where it is read automatically. The only thing you need to do is explicitly enable the mbstring PHP extension, which you can do by typing:

sudo phpenmod mbstring

step 4: Afterwards, restart Apache for your changes to be recognized:

sudo systemctl restart apache2

### step 5:Configuring Password Access for the MySQL Root Account

sudo mysql

Next, check which authentication method each of your MySQL user accounts use with the following command:

SELECT user,authentication\_string,plugin,host FROM mysql.user;

### Step6: To configure the root account to authenticate with a password, run the following ALTER USER command. Be sure to change password to a strong password

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

**Step 7: Then, run FLUSH PRIVILEGES which tells the server to reload the grant tables and put your new changes into effect:**

FLUSH PRIVILEGES;

Step 8: Check the authentication methods employed by each of your users again to confirm that **root** no longer authenticates using the auth\_socket plugin:

SELECT user,authentication\_string,plugin,host FROM mysql.user;

### Step9: Configuring Password Access for a Dedicated MySQL User

mysql -u root -p

step10: From there, create a new user and give it a strong password:

CREATE USER 'sammy'@'localhost' IDENTIFIED BY 'password';

Step11: Then, grant your new user appropriate privileges.

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

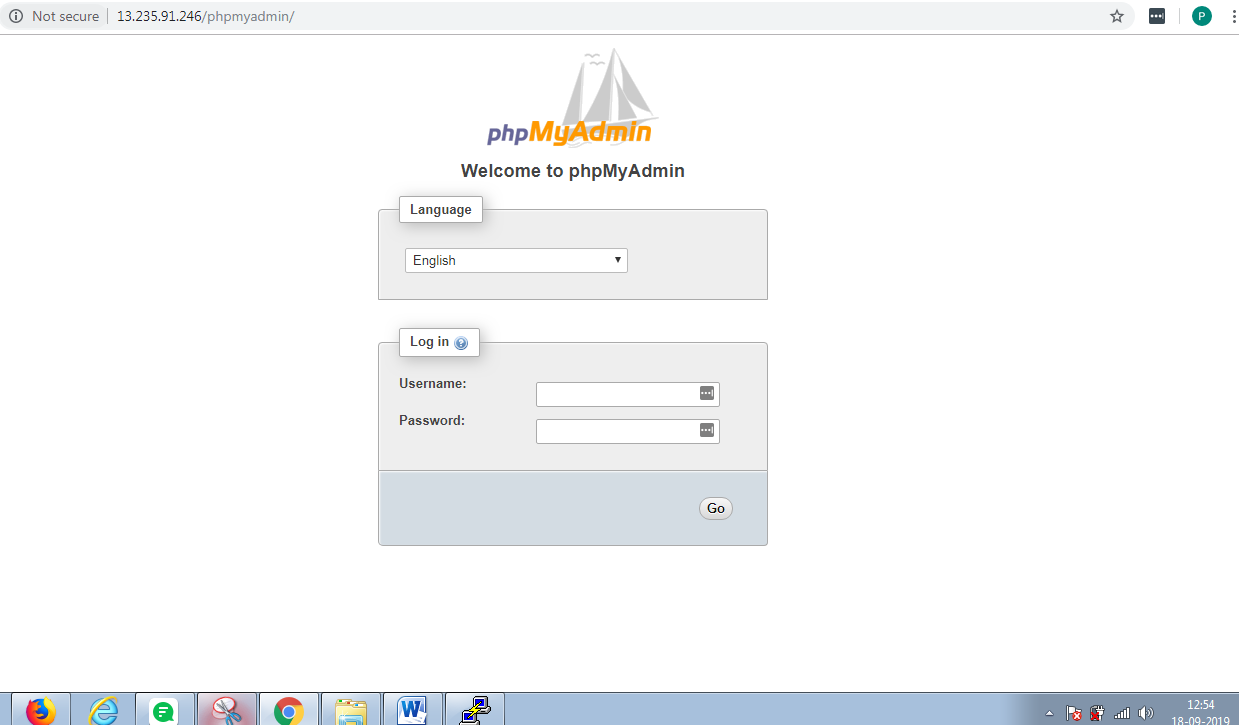
FLUSH PRIVILEGES;

Step12: exit the MySQL shell:

Exit;

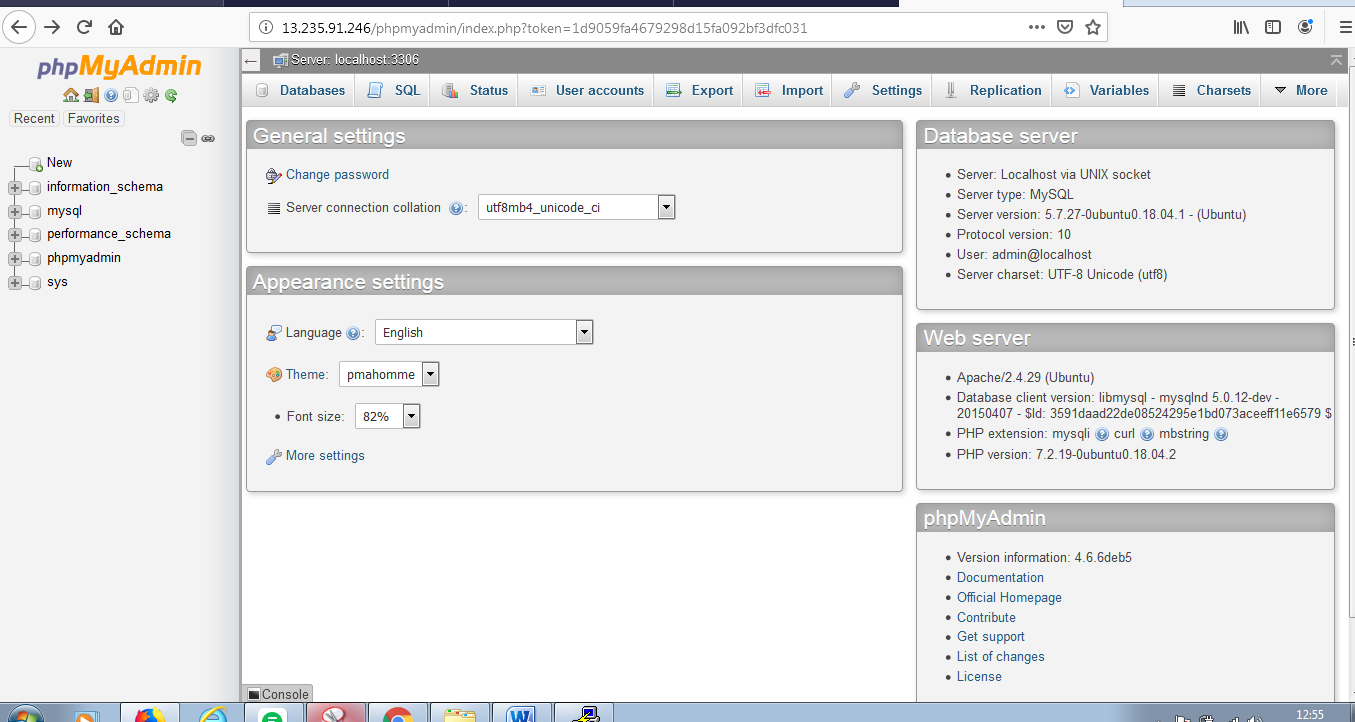
Step 13: You can now access the web interface by visiting your server’s domain name or public IP address followed by /phpmyadmin:

http://your\_domain\_or\_IP/phpmyadmin



Step13: Log in to the interface, either as **root** or with the new username and password you just configured.

When you log in, you’ll see the user interface, which will look something like this:



## Create a Database for WordPress

## Link: https://www.rosehosting.com/blog/how-to-install-wordpress-with-lamp-stack-on-ubuntu-18-04/

Step1: Now, we will create our MySQL database for our WordPress site. Log in to your MySQL server with the following command and enter your MySQL root password:

mysql -u root -p

step2: In this section, we will create a new MySQL database wordpress and assign user access to it to a new user admin\_user with password StrongPassword

CREATE DATABASE wordpress;

GRANT ALL PRIVILEGES ON wordpress.\* TO 'admin\_user'@'localhost' IDENTIFIED BY 'StrongPassword';

FLUSH PRIVILEGES;

exit;

Step3:install wordpress

We can now proceed with the actual installation of WordPress.  Run the following commands to download and extract the latest WordPress installation files in the default web server document root directory (**/var/www/html**).

cd /var/www/html

wget -c http://wordpress.org/latest.zip

unzip latest.zip

chown -R www-data:www-data wordpress

All the WordPress files will be now placed in the **wordpress** directory in **/var/www/html/wordpress**

**Step4:** run the following command to rename the sample configuration file:

cd /var/www/html/wordpress

mv wp-config-sample.php wp-config.php

Step5: Now open the **wp-config.php** file with your favorite text editor, for example:

nano wp-config.php

step6: And update the database settings, replacing ***database\_name\_here***, ***username\_here*** and ***password\_here*** with your own details:

==========================================

// \*\* MySQL settings - You can get this info from your web host \*\* //

/\*\* The name of the database for WordPress \*/

define('DB\_NAME', 'wordpress');

/\*\* MySQL database username \*/

define('DB\_USER', 'admin\_user');

/\*\* MySQL database password \*/

define('DB\_PASSWORD', 'StrongPassword');

/\*\* MySQL hostname \*/

define('DB\_HOST', 'localhost');

/\*\* Database Charset to use in creating database tables. \*/

define('DB\_CHARSET', 'utf8');

/\*\* The Database Collate type. Don't change this if in doubt. \*/

define('DB\_COLLATE', '');

===============================================================

Save and exit the file.

## Step 7: Create the Virtual Host Files

We can now create our virtual host files. Run the following command to create the virtual host configuration file for your domain, **your\_domain.com**:

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

step 8:And add the following content to the file:

<VirtualHost \*:80>

ServerAdmin admin@your\_domain.com

ServerName your\_domain.com

ServerAlias www.your\_domain.com

DocumentRoot /var/www/html/wordpress

<Directory /var/www/html/wordpress>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/your\_domain.com\_error.log

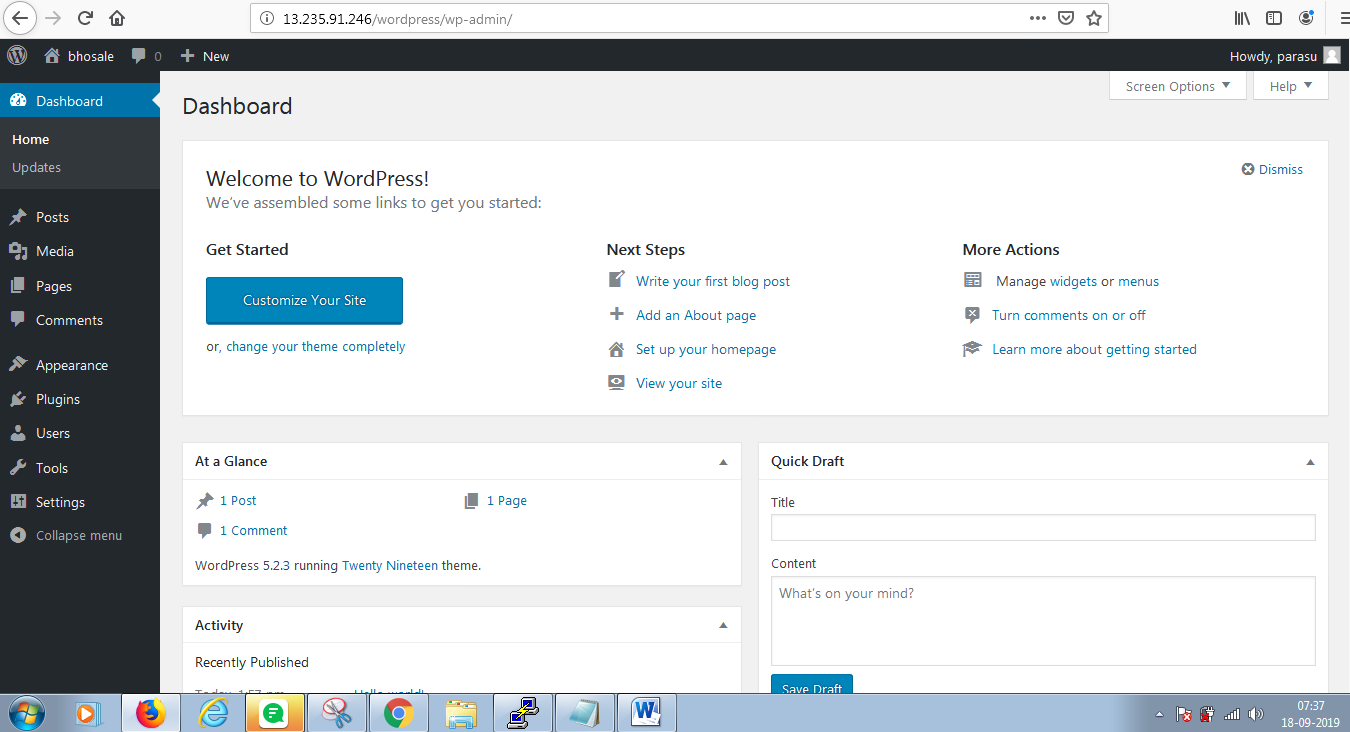
CustomLog ${APACHE\_LOG\_DIR}/your\_domain.com\_access.log combined

</VirtualHost>

Step 9:To enable the virtual host we have just created, run the following command:

ln -s /etc/apache2/sites-available/your\_domain.com.conf /etc/apache2/sites-enabled/your\_domain.com.conf

<http://13.235.91.246/wordpress/wp-admin/>



## Multiple wordpress

## <https://hostpresto.com/community/tutorials/how-to-deploy-multiple-wordpress-sites-on-a-single-ubuntu-16-04-host/>

## https://www.linode.com/docs/websites/cms/configure-apache-to-run-multiple-wordpress-sites-on-one-linode/