Setting up a LAMP(Linux Apache MySQL PHP) server with WordPress

PUBLISHED ON *February 21, 2018 by Mohd Naeem*

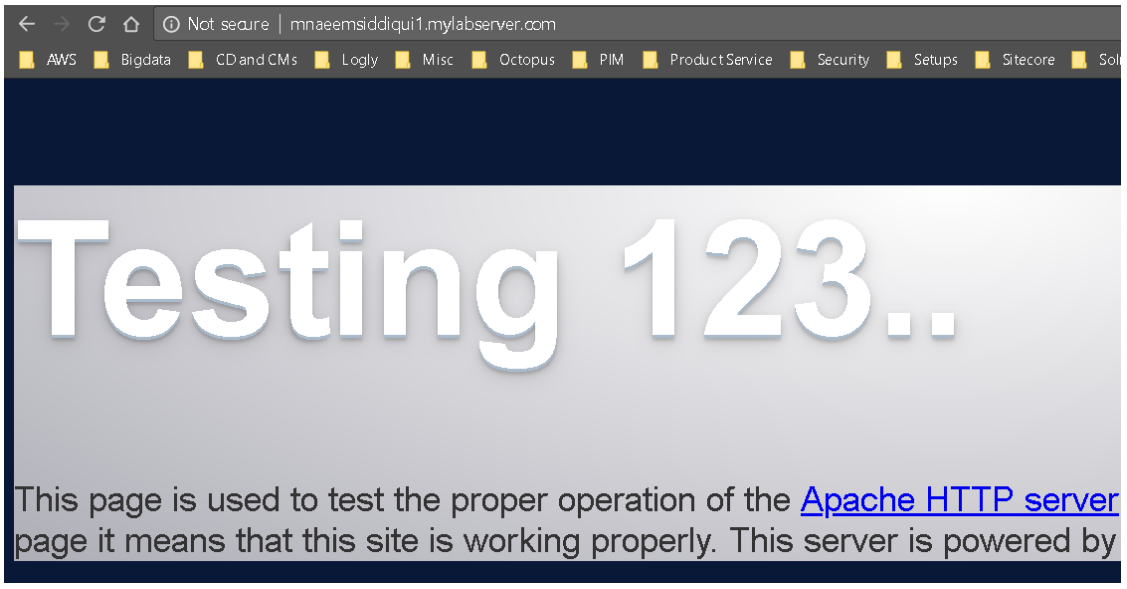
- As you can see this even this site on which I am writing my blogs is also powered on WordPress.
- It gives you a medium to write down your experiences as blogs and posts and lets other like, dislike, comment on your post and you can efficiently transform all this on mobile device too.
- Setting up WordPress on Linux servers is pretty easy and documented all over the Internet and you should not spend thousands on websites, if you can do that for free or almost free(counting on the cost for registering a domain name).
- Lets break down the task on how to setup all this:
  - One, setup up the LAMP stack
  - Two, setup up the WordPress website.
- **Setting up the LAMP server** :
  - We will use CentOS 7 as the OS, but you can use any Linux Distro.
  - The commands would be almost be same except on how to use the Installation managers 'apt' or 'yum' or on how to start services.
  - I will highlight where ever needed.
  - If you have not installed the OS, you can check my one the posts on how to install Linux.
  - So lets **start**:
    - Login to the server and elevate your rights to superuser/root – **sudo su –**
    - Install the required software. You can install Apache, MariaDB, PHP all one by one or all together, I will explain both.
    - First disable SELinux – **setenforce 0**
    - Lets see all together – **yum clean all && yum -y update && yum -y install httpd mariadb mariadb-server php php-common php-mysql php-gd php-xml php-mbstring php-mcrypt php-xmlrpc unzip wget firewalld**
      - If you have **Ubuntu or an Debian Linux based distro**, then use apt-get instead of yum and also some of the names will change e.g. httpd will be replaced by apache2
        - **apt-get clean all && apt-get -y update && apt-get -y install apache2 mysql mysql-server php php-common php-mysql php-gd php-xml php-mbstring php-mcrypt php-xmlrpc unzip wget**
      - You can do one by one installation-
        - **yum -y update** – to update the server
        - **yum -y install httpd –** to install Apache
        - **yum -y install mariadb mariadb-server –** to install MySQL and its dependencies.
        - **yum -y install php php-common php-mysql php-gd php-xml php-mbstring php-mcrypt php-xmlrpc –** to install PHP and related dependencies
        - **yum -y unzip curl wget firewalld –** to install miscellaneous software.

```
mnaeemsiddiqui1 login: user
Password:
You are required to change your password immediately (root enforced)
Changing password for user.
(current) UNIX password:
New password:
Retype new password:
[user@mnaeemsiddiqui1 ~]$ sudo su -
[sudo] password for user:
[root@mnaeemsiddiqui1 ~]# whoami && pwd
root
/root
[root@mnaeemsiddiqui1 ~]# yum clean all && yum -y update && yum -y install httpd mariadb mariadb-server php php-common php-mysql php-gd php-xml ph
p-mbstring php-mcrypt php-xmlrpc unzip wget
Loaded plugins: fastestmirror
Cleaning repos: base epel extras nux-dextop tigervnc-el7 updates xrdp
Cleaning up everything
Maybe you want: rm -rf /var/cache/yum, to also free up space taken by orphaned data from disabled or removed repos
Cleaning up list of fastest mirrors
Loaded plugins: fastestmirror
base                                                                                | 3.6 kB  00:00:00
epel/x86_64/metalink                                                                |  15 kB  00:00:00
epel                                                                                | 4.7 kB  00:00:00
extras                                                                              | 3.4 kB  00:00:00
nux-dextop                                                                          | 2.9 kB  00:00:00
tigervnc-el7                                                                        | 2.9 kB  00:00:00
updates                                                                             | 3.4 kB  00:00:00
xrdp                                                                                | 2.9 kB  00:00:00
(1/10): epel/x86_64/group_gz                                                        | 266 kB  00:00:00
(2/10): base/7/x86_64/group_gz                                                      | 156 kB  00:00:00
(3/10): epel/x86_64/updateinfo                                                      | 892 kB  00:00:00
(4/10): extras/7/x86_64/primary_db                                                  | 166 kB  00:00:00
(5/10): epel/x86_64/primary_db                                                      | 6.2 MB  00:00:00
```

- Now that we install required software, we need to start some services and enable them so that they are always execute on boot.
- **Enable and start Apache –**
  - **systemctl start httpd** (for Ubuntu based systems use – **service apache2 start**)
  - **systemctl enable httpd**(for Ubuntu based systems use – **service apache2 enable**)
- **Enable and start MySQL-**
  - **systemctl start mariadb**(for Ubuntu based systems use – **service mysqld start**)
  - **systemctl enable mariadb**(for Ubuntu based systems use – **service mysqld enable**)
- **Enable and start Firewall-**
  - **systemctl start firewalld**(for Ubuntu based systems use – **service firewalld start**)
  - **systemctl enable httpd**(for Ubuntu based systems use – **service firewalld enable**)

```
[root@mnaeemsiddiqui1 ~]# systemctl start httpd
[root@mnaeemsiddiqui1 ~]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@mnaeemsiddiqui1 ~]#
[root@mnaeemsiddiqui1 ~]# systemctl start mariadb
[root@mnaeemsiddiqui1 ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
[root@mnaeemsiddiqui1 ~]#
[root@mnaeemsiddiqui1 ~]# systemctl start firewalld
[root@mnaeemsiddiqui1 ~]#
```

- Enable Http and Https on the server
  - **firewall-cmd –permanent –zone=public –add-service=http**
  - **firewall-cmd –permanent –zone=public –add-service=https**
  - **firewall-cmd –reload**
  - **systemctl restart httpd**
- Test if the Apache is working or not
  - To check the host name – **hostname or ip addr**
  - Now in browser type : **http://thevalue_ugot_from_hostname_command (http://thevalue_ugot_from_hostname_command)** or **http://ip_addr (http://ip_addr)** or **http://localhost (http://localhost)**
  - If you are getting the below screen then



- Now that Apache is working and also that we have already installed MariaDB, we need to configure and run it
- To configure MySQL or MariaDB run – **mysql_secure_installation**
  - It will start a wizard and will prompt you for password, simply press "Enter" because you have not setup the password as yet
  - Then it will ask – **Set root password? , Type 'Y' and then provide password you want to set.**
  - After that it will prompt you with 4 more questions – **Just type 'Y'** for all, till you get a success message

```
[root@mnaeemsiddiqui1 ~]#
[root@mnaeemsiddiqui1 ~]# mysql_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
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

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

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

Set root password? [Y/n] y
New password:
Re-enter new password:
Sorry, passwords do not match.

New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!


By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.
```

  - Now login to MySQL – **mysql -u root -p**
  - Use the above password and login to get the MariaDB prompt.

- Now we have to create the following :
  - a **database** for WordPress – **create database mywordpress;**
  - a **user** for WordPress – **create user mywpuser@localhost identified by 'mywppass';**
  - grant all privileges to this user – **grant all privileges on mywordpress.* to mywpuser@localhost;**
  - flush the privileges so that they get applied – **flush privileges;**
  - then exit MySQL – **exit;**

```
[root@mnaeemsiddiqui1 ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 5.5.56-MariaDB MariaDB Server

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database mywordpress;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> create user mywpuser@localhost identified by 'mywppass';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> grant all privileges on mywordpress.* to mywpuser@localhost
    -> ;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit;
Bye
[root@mnaeemsiddiqui1 ~]#
```

- Set ReWriteRule to :
  - edit **/etc/httpd/conf/httpd.conf**
  - and update **AllowOverride All**

```
 GNU nano 2.3.1                          File: /etc/httpd/conf/httpd.conf

# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
#


#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed.  This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
#
ServerAdmin root@localhost

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
#ServerName www.example.com:80

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride All
    Require all denied
```

- **Setting Up WordPress:**
  - Download latest WordPress package – **wget https://wordpress.org/latest.zip (https://wordpress.org/latest.zip)**
  - Unzip the zip file – **unzip latest.zip**

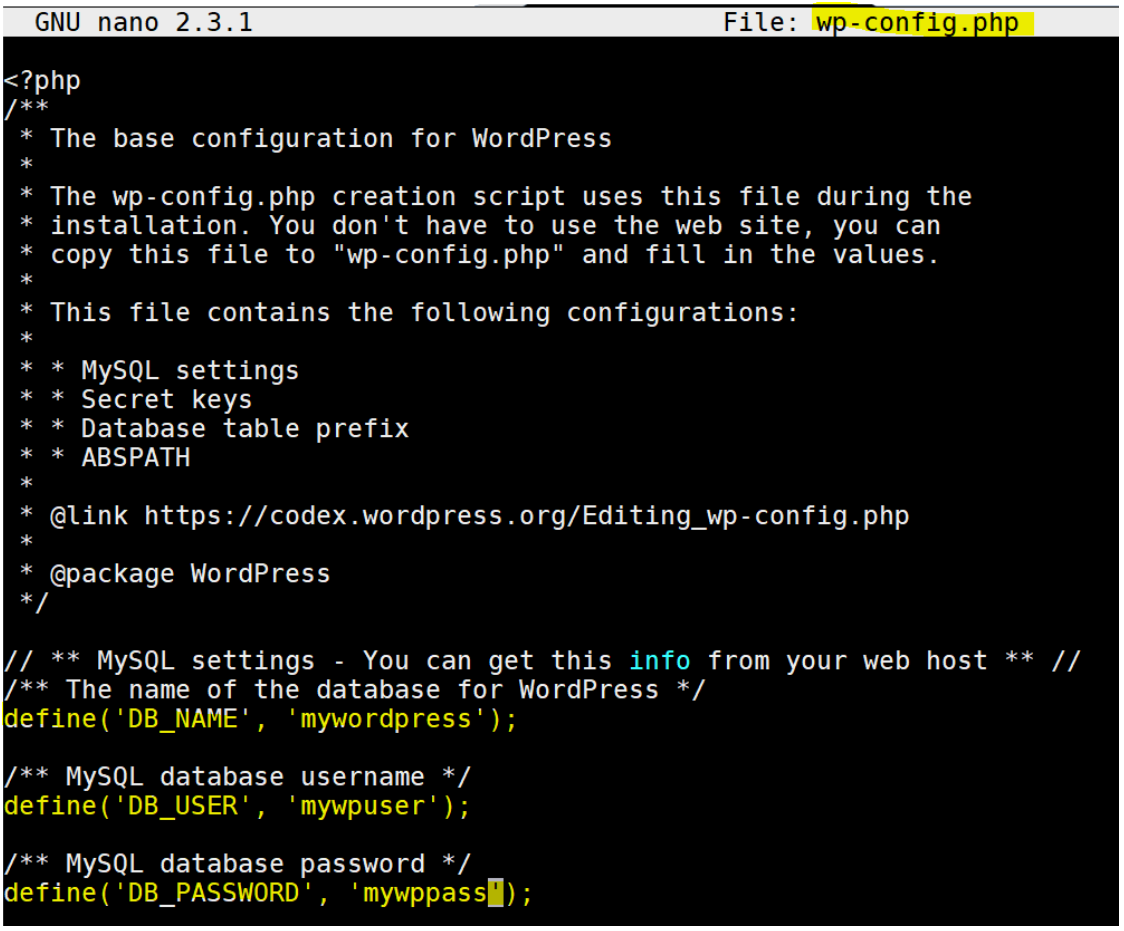
```
[root@mnaeemsiddiqui1 ~]# wget https://wordpress.org/latest.zip
--2018-02-22 03:49:00--  https://wordpress.org/latest.zip
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9332728 (8.9M) [application/zip]
Saving to: 'latest.zip'

100%[======================================================================================

2018-02-22 03:49:01 (20.0 MB/s) - 'latest.zip' saved [9332728/9332728]

[root@mnaeemsiddiqui1 ~]# 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/
   creating: wordpress/wp-content/themes/twentysixteen/
  inflating: wordpress/wp-content/themes/twentysixteen/rtl.css
  inflating: wordpress/wp-content/themes/twentysixteen/footer.php
  inflating: wordpress/wp-content/themes/twentysixteen/sidebar.php
   creating: wordpress/wp-content/themes/twentysixteen/template-parts/
  inflating: wordpress/wp-content/themes/twentysixteen/template-parts/content-page.php
  inflating: wordpress/wp-content/themes/twentysixteen/template-parts/content-none.php
```

  - View the wordpress folder created by the above action – **ls -ltr wordpress**
  - Now copy the wordpress folder to /var/www/html – **cp -avr wordpress /var/www/html**
  - Grant permissions on wordpress folder –

- **chmod -R 755 /var/www/html/**
- Change the owner pf wordpress folder to Apache –
  - **chown -R apache:apache /var/www/html/**
- Rename file 'wp-config-sample.php' to 'wp-config.php'- **mv wp-config-sample.php wp-config.php**

```
[root@mnaeemsiddiqui1 html]# chmod -R 775 wordpress
[root@mnaeemsiddiqui1 html]# chown -R apache:apache wordpress
[root@mnaeemsiddiqui1 html]# ls -ltr wordpress
total 184
-rwxrwxr-x.  1 apache apache   418 Sep 25  2013 index.php
-rwxrwxr-x.  1 apache apache  2853 Dec 16  2015 wp-config-sample.php
-rwxrwxr-x.  1 apache apache   364 Dec 19  2015 wp-blog-header.php
-rwxrwxr-x.  1 apache apache  1627 Aug 29  2016 wp-comments-post.php
-rwxrwxr-x.  1 apache apache  3065 Aug 31  2016 xmlrpc.php
-rwxrwxr-x.  1 apache apache  2422 Nov 21  2016 wp-links-opml.php
-rwxrwxr-x.  1 apache apache  7413 Dec 12  2016 readme.html
-rwxrwxr-x.  1 apache apache  8048 Jan 11  2017 wp-mail.php
-rwxrwxr-x.  1 apache apache  3669 Aug 20  2017 wp-cron.php
-rwxrwxr-x.  1 apache apache  3306 Aug 22  2017 wp-load.php
-rwxrwxr-x.  1 apache apache  5434 Sep 23 12:21 wp-activate.php
-rwxrwxr-x.  1 apache apache 16246 Oct  4 00:20 wp-settings.php
-rwxrwxr-x.  1 apache apache 36583 Oct 13 02:10 wp-login.php
-rwxrwxr-x.  1 apache apache 30071 Oct 18 17:36 wp-signup.php
-rwxrwxr-x.  1 apache apache  4620 Oct 23 22:12 wp-trackback.php
-rwxrwxr-x.  1 apache apache 19935 Jan  6 19:32 license.txt
drwxrwxr-x. 18 apache apache  8192 Feb  6 15:49 wp-includes
drwxrwxr-x.  4 apache apache    49 Feb  6 15:49 wp-content
drwxrwxr-x.  9 apache apache  4096 Feb  6 15:49 wp-admin
[root@mnaeemsiddiqui1 html]# cd wordpress
[root@mnaeemsiddiqui1 wordpress]# ls -ltr wp-config-sample.php
-rwxrwxr-x. 1 apache apache 2853 Dec 16  2015 wp-config-sample.php
[root@mnaeemsiddiqui1 wordpress]# mv wp-config-sample.php wp-config.php
[root@mnaeemsiddiqui1 wordpress]# nano wp-config-php
```

- Edit file wp-config.php – **nano wp-config.php** with follwing details:
  - **define('DB_NAME', 'mywordpress');**
    define('DB_USER', 'mywpuser');
    define('DB_PASSWORD', 'mywppass');



- We are done with installation, lets try connecting to WordPress
  - Type –
    - **http://localhost/wordpress (http://localhost/wordpress)** if your host is local
    - **http://ip_addr/wordpress(get (http://ip_addr/wordpress(get) IP Address by executing ip addr command)**
    - **http://hostname/wordpress (http://hostname/wordpress) (get Host name by executing hostname command)**

WordPress

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title MyWordPress

Username mywpuser001

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password •••••••••• mywppass01 Hide

Medium

Important: You will need this password to log in. Please store it in a secure location.

Your Email test@best.com

Double-check your email address before continuing.

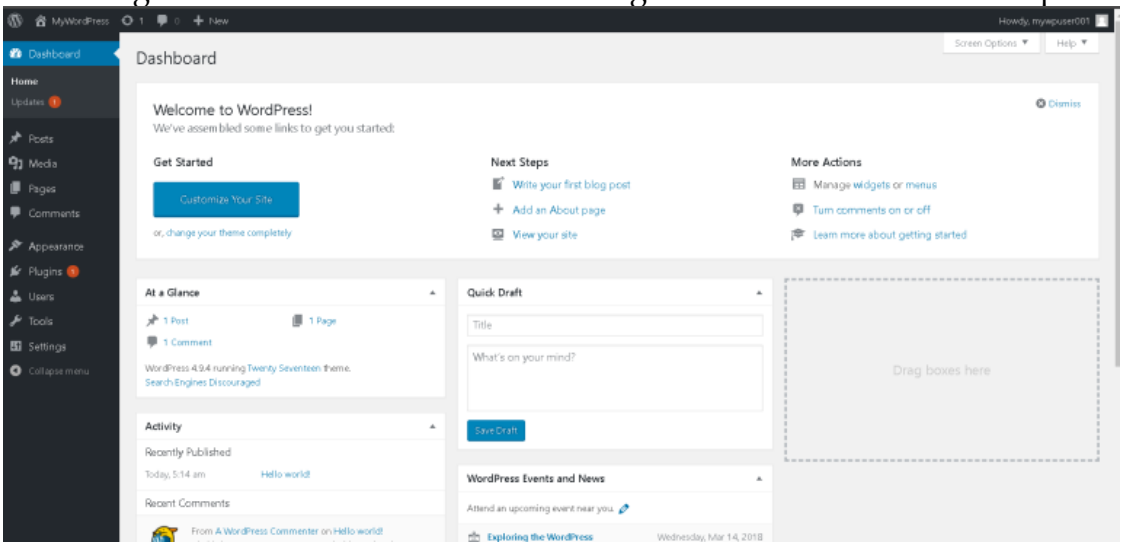
Search Engine Visibility

Search Engine Visibility

Discourage search engines from indexing this site

It is up to search engines to honor this request.

- Install WordPress
- You will get a setup page to enter site name, create a user and then you can click 'Install WordPress'.
- You will get a success screen and a link to login – use the above user and password to login



- yay!!!, you have successfully installed WordPress on a LAMP stack.

CATEGORIES LINUX