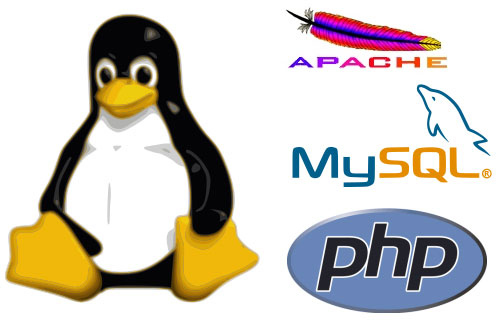
**How To Install LAMP Stack On Ubuntu 16.04**

[](https://s24255.pcdn.co/wp-content/uploads/2014/05/lamp-server.jpg)

**LAMP** is a combination of operating system and open-source software stack. The acronym LAMP came from the first letters of **L**inux, **A**pache HTTP Server, **M**ySQL or **M**ariaDB database, and **P**HP/**P**erl/**P**ython. This tutorial describes how to install LAMP stack on Ubuntu 16.04, however the steps described below should work on previous Ubuntu versions such as Ubuntu 15.10/15.04/14.10/14.04/13.10 etc.

**Install LAMP Stack On Ubuntu 16.04**

1. Install Apache

**Apache** is an open-source multi-platform web server. It provides a full range of web server features including CGI, SSL and virtual domains.

To install Apache, enter:

sudo apt-get install apache2

Enable  and  start   your  apache

root@ubuntu:~# systemctl    enable  apache2

root@ubuntu:~# systemctl    start  apache2

root@ubuntu:~# systemctl    status  apache2

● apache2.service - LSB: Apache2 web server

   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)

   Active: active (running) since Wed 2016-03-02 09:23:37 PST; 1min 2s ago

     Docs: man:systemd-sysv-generator(8)

   CGroup: /system.slice/apache2.service

           ├─22328 /usr/sbin/apache2 -k start

           ├─22331 /usr/sbin/apache2 -k start

           └─22332 /usr/sbin/apache2 -k start

Mar 02 09:23:36 ubuntu systemd[1]: Starting LSB: Apache2 web server...

Mar 02 09:23:36 ubuntu apache2[22304]:  \* Starting web server apache2

Mar 02 09:23:36 ubuntu apache2[22304]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'S

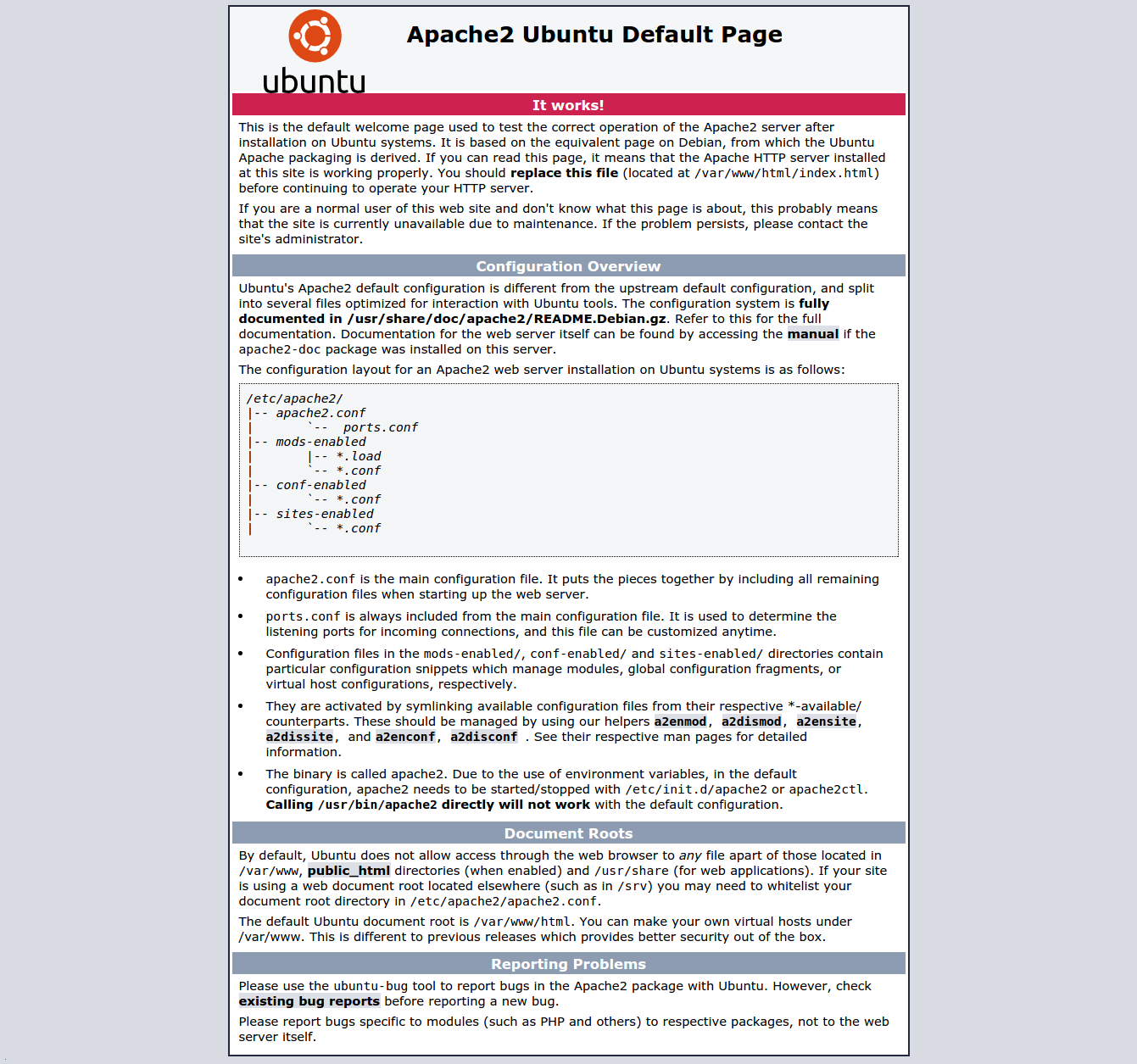
Mar 02 09:23:37 ubuntu apache2[22304]:  \*

Mar 02 09:23:37 ubuntu systemd[1]: Started LSB: Apache2 web server.

Mar 02 09:24:34 ubuntu systemd[1]: Started LSB: Apache2 web server.

Test Apache:

Open your web browser and navigate to **http://localhost/** or **http://server-ip-address/**.

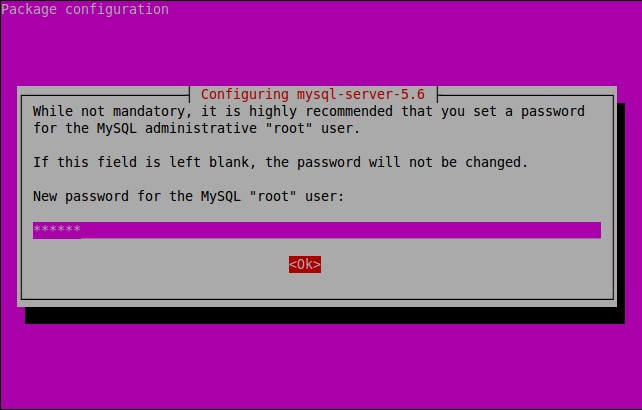
[](https://s24255.pcdn.co/wp-content/uploads/2015/10/Apache2-Ubuntu-Default-Page.png)

2. Install MySQL

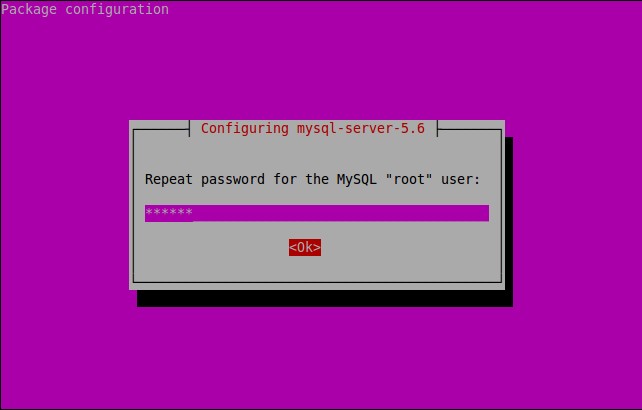
**MySQL** is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments

sudo apt-get install mysql-server mysql-client

During installation, you’ll be asked to setup the MySQL “root” user password. Enter the password and click Ok.

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/sk@server-_0012.jpg)

Re-enter the password.

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/sk@server-_0022.jpg)

MySQL is installed now.

You can verify the MySQL server status using command:

**On Ubuntu 16.04/15.10/15.04:**

sudo systemctl status mysql

**On Ubuntu 14.10 and previous versions:**

sudo service mysql status

Sample output:

● mysql.service - MySQL Community Server

Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)

Active: active (running) since Mon 2015-10-26 14:23:01 IST; 28s ago

Main PID: 3577 (mysqld\_safe)

CGroup: /system.slice/mysql.service

├─3577 /bin/sh /usr/bin/mysqld\_safe

└─3924 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --...

Oct 26 14:23:00 server systemd[1]: Starting MySQL Community Server...

Oct 26 14:23:00 server mysqld\_safe[3577]: 151026 14:23:00 mysqld\_safe Can't ....

Oct 26 14:23:00 server mysqld\_safe[3577]: 151026 14:23:00 mysqld\_safe Loggin....

Oct 26 14:23:01 server mysqld\_safe[3577]: 151026 14:23:01 mysqld\_safe Starti...l

Oct 26 14:23:01 server systemd[1]: Started MySQL Community Server.

Hint: Some lines were ellipsized, use -l to show in full.

3. Install MariaDB

In case you want to use MariaDB instead of MySQL community edition, follow the steps given below.

**MariaDB** is a drop in replacement for MySQL. It is a robust, scalable and reliable SQL server that comes rich set of enhancements.

First you have to remove existing MySQL packages if any. To completely uninstall MySQL along with its configuration files, enter the following commands one by one:

sudo systemctl stop mysql

sudo apt-get remove --purge mysql-server mysql-client mysql-common

sudo apt-get autoremove

sudo apt-get autoclean

sudo rm -rf /var/lib/mysql/

sudo rm -rf /etc/mysql/

After removing MySQL, run the following command to install MariaDB.

sudo apt-get install mariadb-server

Alternatively, you can install it using [MariaDB repository](https://downloads.mariadb.org/mariadb/repositories/#mirror=tsinghua) if you want to try most recent version of MariaDB. Run the following commands to add PPA. As of writing this, MariaDB PPA is not yet updated to Ubuntu 16.04. However, we can use the repository of Ubuntu 15.10 instead.

sudo apt-get install software-properties-common

sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xcbcb082a1bb943db

sudo add-apt-repository 'deb http://download.nus.edu.sg/mirror/mariadb/repo/10.1/ubuntu vivid main'

Update the software sources list and install MariaDB using following commands:

sudo apt-get update

sudo apt-get install mariadb-server

During installation you will be asked to set mysql ‘root’ user password. Enter the password twice, and complete the installation.

**Important Note:**In Ubuntu 16.04/15.10/15.04, MariaDB won’t ask you to set root user password during installation.

Also, you can’t set password manually using the following command too:

mysql\_secure\_installation

It will throw the following error.

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):

**ERROR 1698 (28000): Access denied for user 'root'@'localhost'**

Enter current password for root (enter for none):

To fix this error, log in to mysql prompt without password as root user:

sudo mysql -u root

Then, run the following commands one by one in the mysql prompt:

use mysql;

update user set plugin='' where User='root';

flush privileges;

\q

That’s it. Now, set database administrative “root” user password using command. Press Enter and go with the default settings.

mysql\_secure\_installation

Sample output:

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:

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.

Remove anonymous users? [Y/n]

 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This

ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n]

 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can

access.  This is also intended only for testing, and should be removed

before moving into a production environment.

Remove test database and access to it? [Y/n]

 - Dropping test database...

ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist

 ... Failed!  Not critical, keep moving...

 - Removing privileges on test database...

 ... Success!

Reloading the privilege tables will ensure that all changes made so far

will take effect immediately.

Reload privilege tables now? [Y/n]

 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB

installation should now be secure.

Thanks for using MariaDB!

Check if mariadb is running or not, using the following command:

sudo service mysql status

Sample output:

● mysql.service - LSB: Start and stop the mysql database server daemon

**Loaded: loaded (/etc/init.d/mysql)**

**Active: active (running) since Mon 2015-10-26 14:26:36 IST; 2min 5s ago**

Docs: man:systemd-sysv-generator(8)

CGroup: /system.slice/mysql.service

├─6387 /bin/bash /usr/bin/mysqld\_safe

├─6388 logger -p daemon err -t /etc/init.d/mysql -i

└─6544 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --...

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: mysql.time\_zone\_transit...

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: mysql.time\_zone\_transit...

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: mysql.user ...

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: Phase 2/6: Fixing views

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: Processing databases

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: information\_schema

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: mysql

Oct 26 14:26:39 server /etc/mysql/debian-start[6595]: performance\_schema

Oct 26 14:26:39 server /etc/mysql/debian-start[6653]: Checking for insecure r...

Oct 26 14:26:39 server /etc/mysql/debian-start[6657]: Triggering myisam-recov...

Hint: Some lines were ellipsized, use -l to show in full.

4. Install PHP  (PHP7)

**PHP** (recursive acronym for PHP: Hypertext Preprocessor) is a widely used open-source general purpose scripting language that is especially suited for web development and can be embedded into HTML.

Install PHP with following command:

sudo apt-get update

sudo apt-get install php7.0-mysql php7.0-curl php7.0-json php7.0-cgi  php7.0 libapache2-mod-php7

Test  your php version

root@ubuntu:~# php -v

PHP 7.0.3-3 (cli) ( NTS )

Copyright (c) 1997-2016 The PHP Group

Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies

    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies

root@ubuntu:~#

To test PHP, create a sample “testphp.php” file in Apache document root folder.

sudo vi /var/www/html/testphp.php

Add the following lines:

<?php

phpinfo();

?>

Restart apache2 service.

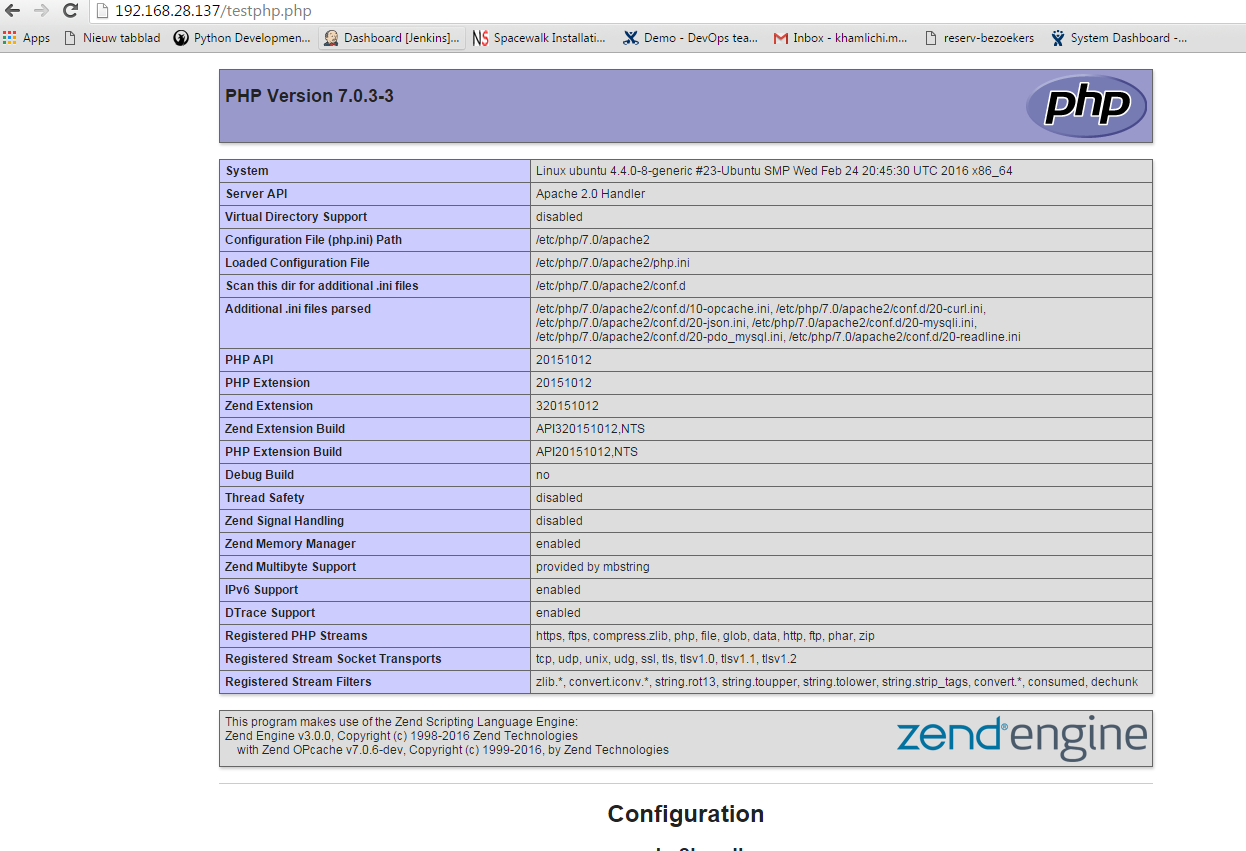
On Ubuntu 15.10/15.04:

sudo systemctl restart apache2

On Ubuntu 14.10 and lower versions:

sudo service apache2 restart

Navigate to **http://server-ip-address/testphp.php**. It will display all the details about php such as version, build date and commands etc.



If you want to install all php modules at once, enter the command **sudo apt-get install php\*** and restart the apache2 service. To verify the modules, open web browser and navigate to **http://server-ip-address/testphp.php**. You will able to see all installed php modules.

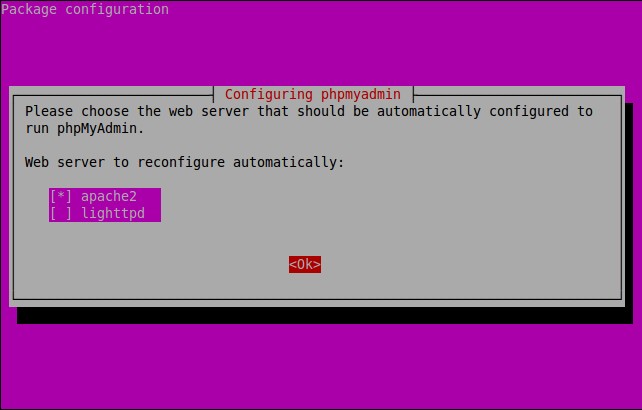
5. Manage MySQL Databases (Optional)

Install phpMyAdmin

**phpMyAdmin** is a free open-source web interface tool used to manage your MySQL databases. It is available in the Official Debian repositories. So install it with command:

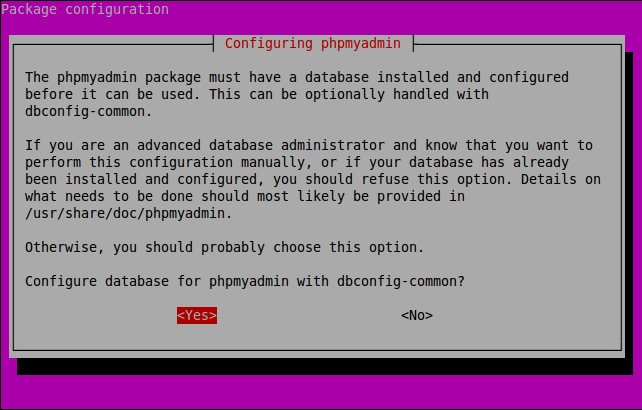
sudo apt-get install phpmyadmin

Select the Web server that should be automatically configured to run phpMyAdmin. In my case, it is apache2.

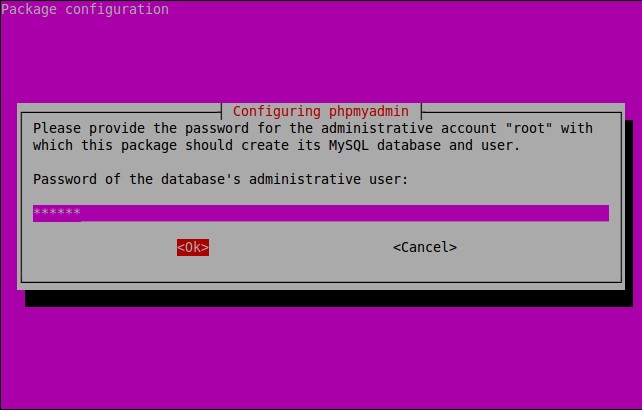
[](https://s24255.pcdn.co/wp-content/uploads/2015/10/sk@server-_004.jpg)

The phpMyAdmin must have a database installed and configured before it can be used. This can be optionally handled by dbconfig-common.

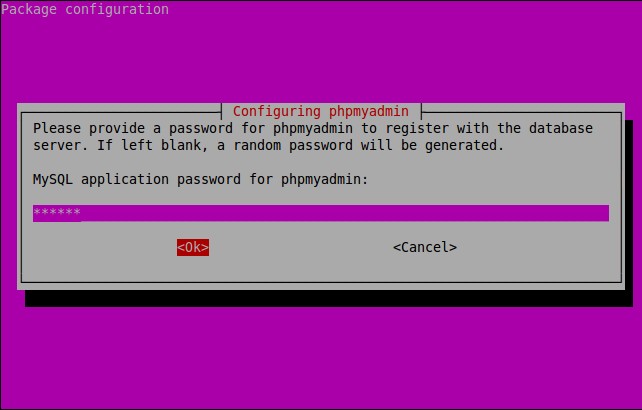
Select ‘Yes’ to configure database for phpmyadmin wjth dbconfig-common.

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/sk@server-_005.jpg)

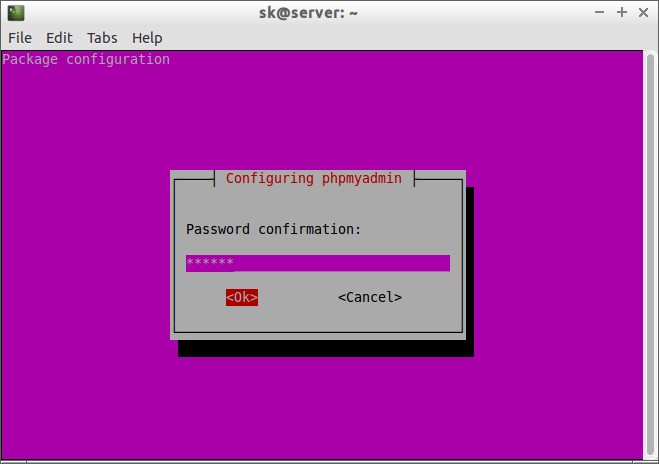
Enter password of the database’s administrative user.

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/sk@server-_008.jpg)

Enter MySQL application password for phpmyadmin:

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/sk@server-_006.jpg)

Re-enter password:

[](https://s24255.pcdn.co/wp-content/uploads/2015/04/sk@server-_009.jpg)

Success! phpMyAdmin installation is installed.

**Additional Note:** if you followed all steps carefully, phpMyAdmin should work just fine. In case phpMyAdmin is not working, please do the following steps.

Open terminal, and type:

sudo nano /etc/apache2/apache2.conf

Add the following line at the end.

Include /etc/phpmyadmin/apache.conf

Save and Exit. Restart apache service:

**On Ubuntu 16.04/15.10/15.04:**

sudo systemctl restart apache2

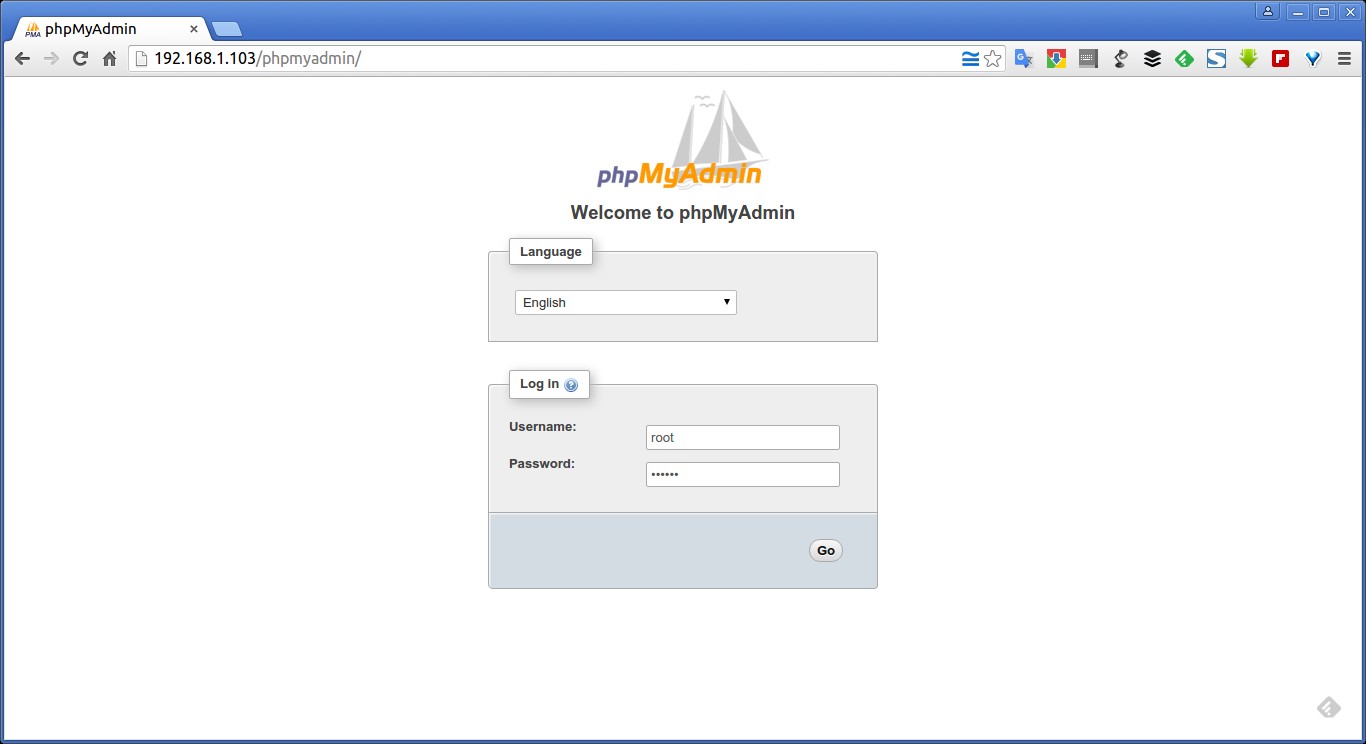
**On Ubuntu 14.10 and lower versions:**

sudo /etc/init.d/apache2 restart

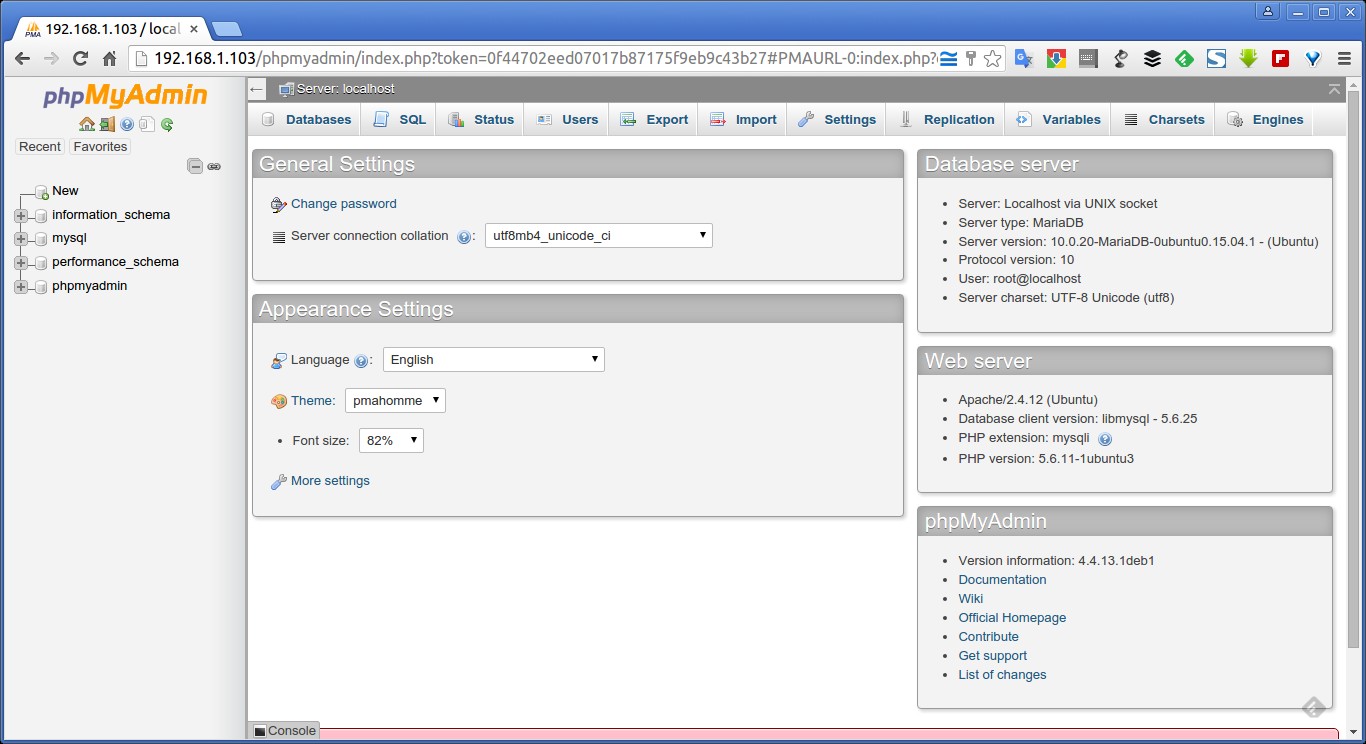
6. Access phpMyAdmin Web Console

Now, you can access the phpmyadmin console by navigating to **http://server-ip-address/phpmyadmin/** from your browser.

Enter your MySQL username and password which you have given in previous steps. In my case its “root” and “ubuntu”.

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/phpMyAdmin-Google-Chrome_009.jpg)

You will be redirected to PhpMyAdmin main web interface.

[](https://s24255.pcdn.co/wp-content/uploads/2015/10/192.168.1.103-localhost-phpMyAdmin-4.4.13.1deb1-Google-Chrome_010.jpg)

From here, you can manage your MySQL databases from phpMyAdmin web interface.

That’s it. Your LAMP stack is ready to use.