TecMint
The Ideal Linux Blog for Sysadmins & Geeks

How to Install LAMP Stack with PhpMyAdmin in Ubuntu 20.04

LAMP stack is the combination of the most frequently used software packages to build dynamic websites. **LAMP** is an abbreviation that uses the first letter of each of the packages included in it: **L**inux, **A**pache, **M**ariaDB, and **P**HP.

You can use **LAMP** to build awesome websites with platforms such as WordPress or Joomla for example.

Additionally, by default, **MySQL/MariaDB** databases are managed from the command-line interface, via the MySQL shell. If you prefer to manage your databases and perform other useful database server operations from a graphical interface, you need to install **PhpMyAdmin**, a popular PHP-based web application.

If you looking for a LAMP setup for your Ubuntu 20.04, then you should read our LEMP setup guide on Ubuntu 20.04.

In this article, you will learn how to install and configure **LAMP** with **PhpMyAdmin** in **Ubuntu 20.04** server. The guide assumes that you have already installed **Ubuntu 20.04**. If you have not installed already, you can refer to our guides here:

Prerequisites:

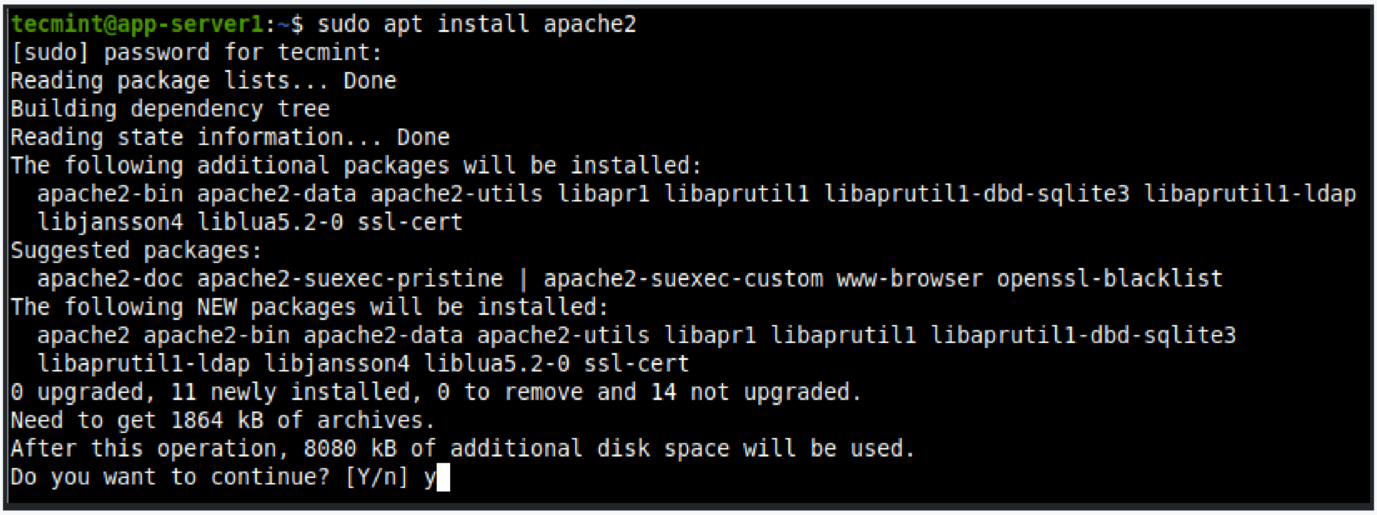
1. Ubuntu 20.04 Server Installation Guide

Step 1: Installing Apache on Ubuntu 20.04

1. Apache2 is an open-source popular, powerful, reliable, and high extensible web/HTTP server software used by numerous websites on the internet.

To install the **Apache2** package, use the default package manager as follows:

```
$ sudo apt install apache2
```



```
tecmint@app-server1:~$ sudo apt install apache2
[sudo] password for tecmint:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  libjansson4 liblua5.2-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libjansson4 liblua5.2-0 ssl-cert
0 upgraded, 11 newly installed, 0 to remove and 14 not upgraded.
Need to get 1864 kB of archives.
After this operation, 8080 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

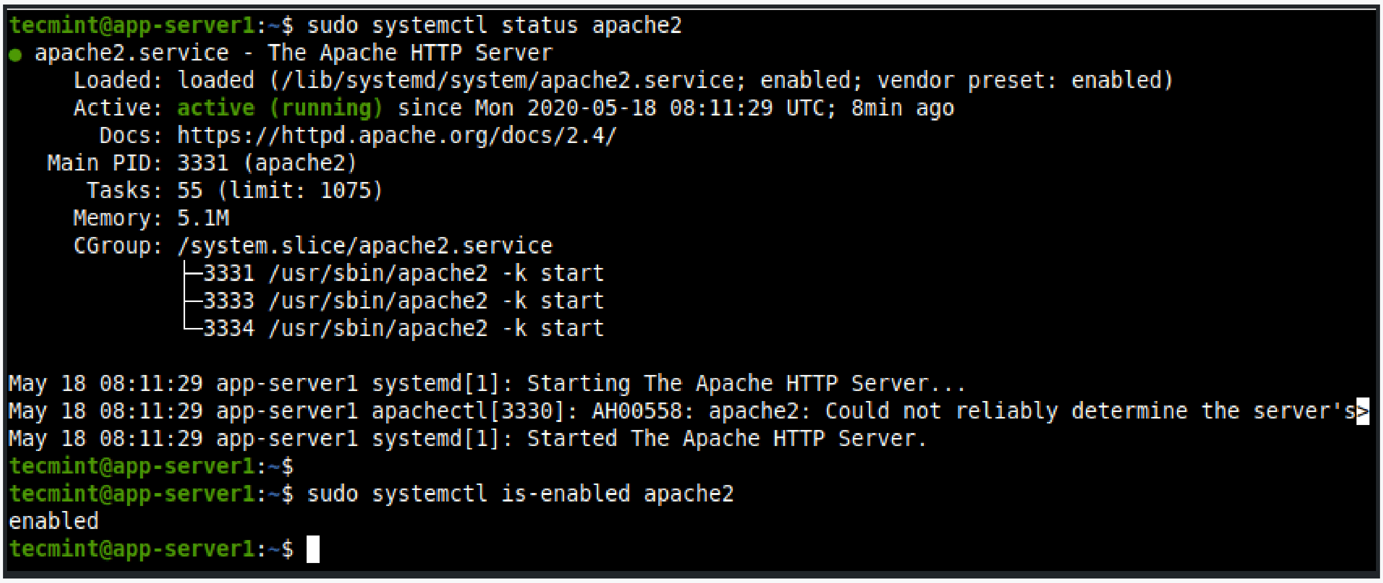
Install Apache on Ubuntu 20.04

The configuration files for **Apache2** are located in **/etc/apache2** directory and the main configuration file is **/etc//etc/apache2/apache2.conf**. And the default document root for storing your web files is **/var/www/html/**.

2. On **Ubuntu** unlike on other major Linux distributions, **systemd** services are automatically started and enabled to start at system boot, when a package (intended to run as a service) installation is complete.

You can confirm that the **Apache2** service is up and enabled on boot using the following systemctl commands.

```
$ sudo systemctl status apache2
$ sudo systemctl is-enabled apache2
```



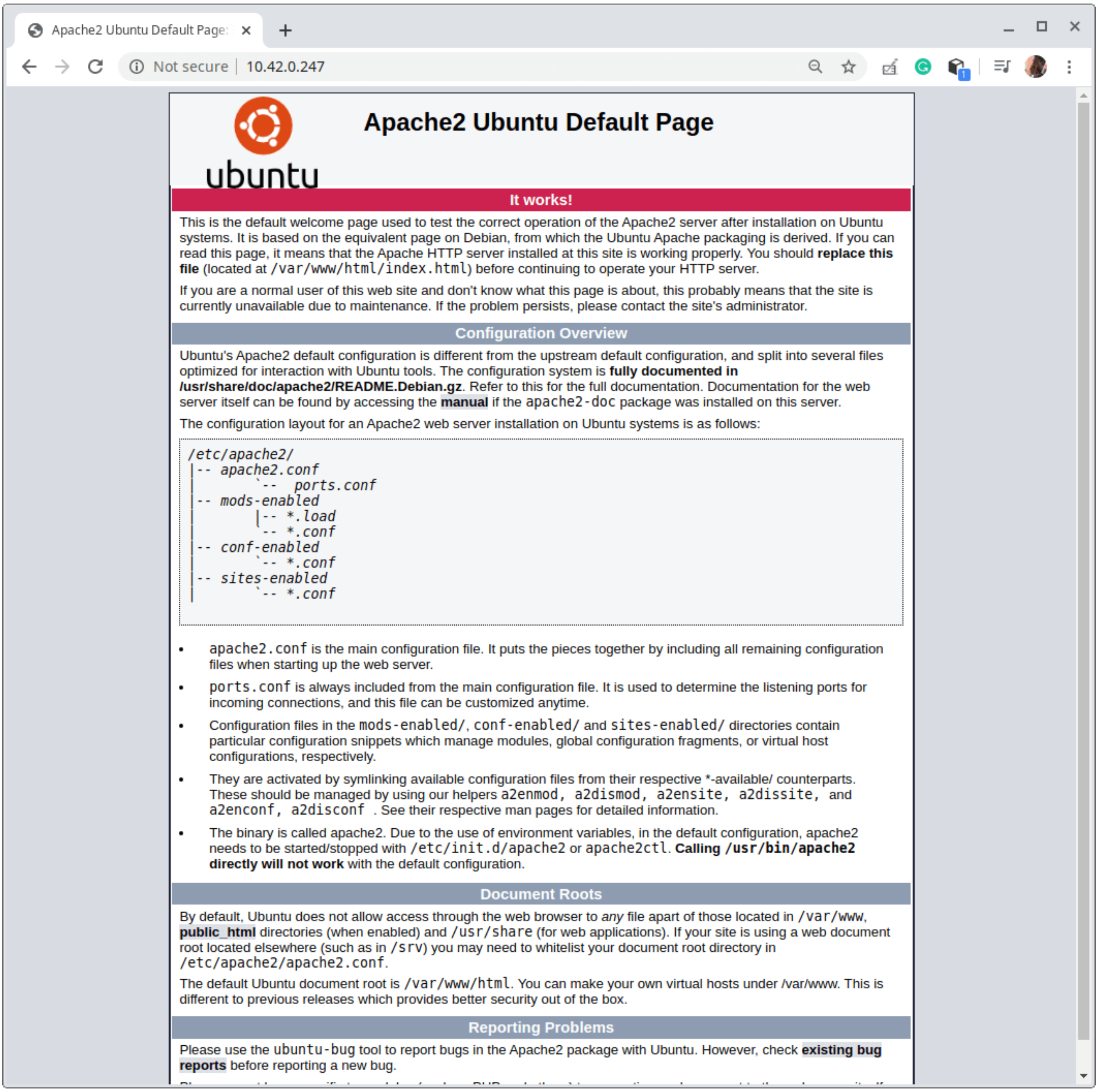
```
tecmint@app-server1:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2020-05-18 08:11:29 UTC; 8min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 3331 (apache2)
      Tasks: 55 (limit: 1075)
     Memory: 5.1M
     CGroup: /system.slice/apache2.service
             ├─3331 /usr/sbin/apache2 -k start
             ├─3333 /usr/sbin/apache2 -k start
             └─3334 /usr/sbin/apache2 -k start

May 18 08:11:29 app-server1 systemd[1]: Starting The Apache HTTP Server...
May 18 08:11:29 app-server1 apachectl[3330]: AH00558: apache2: Could not reliably determine the server's>
May 18 08:11:29 app-server1 systemd[1]: Started The Apache HTTP Server.
tecmint@app-server1:~$
tecmint@app-server1:~$ sudo systemctl is-enabled apache2
enabled
tecmint@app-server1:~$
```

Check Apache Service

4. Next, you need to test the correct operation of the **Apache2** server installation. Open a web browser and use the following address to navigate.

```
http://YOUR_SERVER_IP
```

You should see the Apache Ubuntu default page shown in the screenshot.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default configuration, apache2 needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under /var/www. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

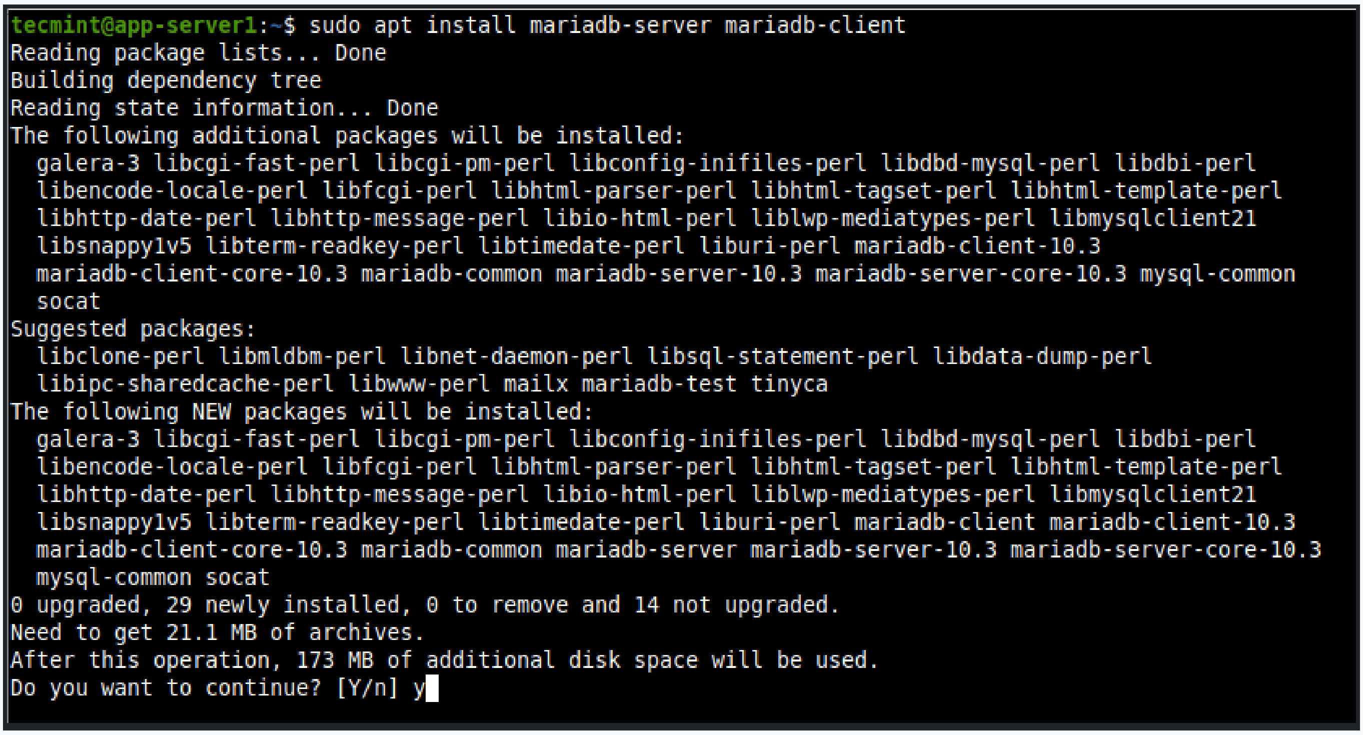
Check Apache Default Page

Step 2: Installing MariaDB Database on Ubuntu 20.04

5. MariaDB is a fork of the popular **MySQL** database. It is now popular too and is the default in most Linux distributions including **Ubuntu** and is also part of most cloud offerings.

To install the **MariaDB** database server and client, run the following command.

```
$ sudo apt install mariadb-server mariadb-client
```



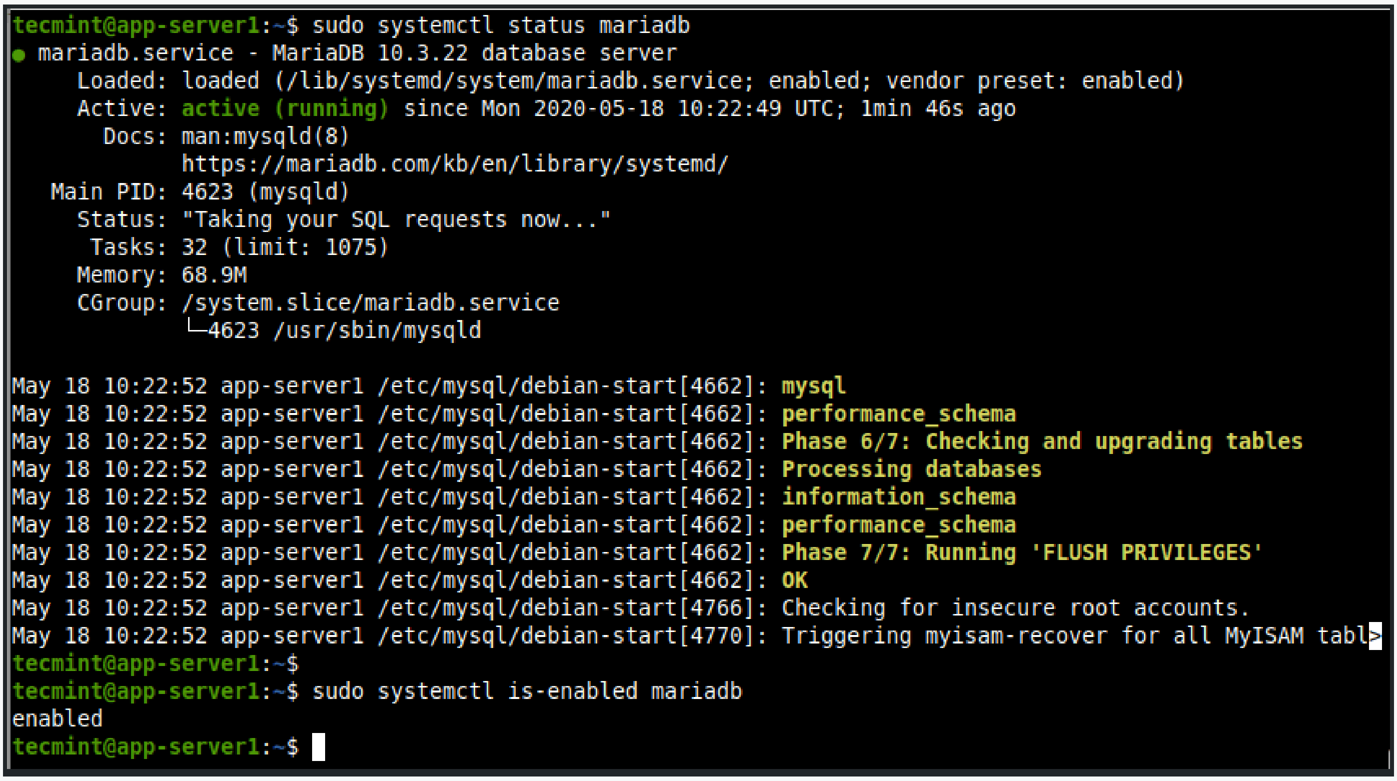
```
tecmint@app-server1:~$ sudo apt install mariadb-server mariadb-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  galera-3 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmysqlclient21
  libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl mariadb-client-10.3
  mariadb-client-core-10.3 mariadb-common mariadb-server-10.3 mariadb-server-core-10.3 mysql-common
  socat
Suggested packages:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl
  libipc-sharedcache-perl libwww-perl mailx mariadb-test tinyca
The following NEW packages will be installed:
  galera-3 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmysqlclient21
  libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl mariadb-client mariadb-client-10.3
  mariadb-client-core-10.3 mariadb-common mariadb-server mariadb-server-10.3 mariadb-server-core-10.3
  mysql-common socat
0 upgraded, 29 newly installed, 0 to remove and 14 not upgraded.
Need to get 21.1 MB of archives.
After this operation, 173 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Install MariaDB on Ubuntu 20.04

The MariaDB configuration files are stored under the **/etc/mysql/** directory. There are so many configuration files in there, you can read the MariaDB documentation for more information.

6. Next, confirm that the **MariaDB** database service is running and is enabled to automatically start when your system is restarted.

```
$ sudo systemctl status mariadb
$ sudo systemctl is-enabled mariadb
```



```
tecmint@app-server1:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.3.22 database server
     Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2020-05-18 10:22:49 UTC; 1min 46s ago
       Docs: man:mysqld(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 4623 (mysqld)
     Status: "Taking your SQL requests now..."
      Tasks: 32 (limit: 1075)
     Memory: 68.9M
     CGroup: /system.slice/mariadb.service
             └─4623 /usr/sbin/mysqld

May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: mysql
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: performance_schema
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: Phase 6/7: Checking and upgrading tables
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: Processing databases
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: information_schema
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: performance_schema
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: Phase 7/7: Running 'FLUSH PRIVILEGES'
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4662]: OK
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4766]: Checking for insecure root accounts.
May 18 10:22:52 app-server1 /etc/mysql/debian-start[4770]: Triggering myisam-recover for all MyISAM tabl>
tecmint@app-server1:~$
tecmint@app-server1:~$ sudo systemctl is-enabled mariadb
enabled
tecmint@app-server1:~$ 
```

Check MariaDB Service

7. On production servers, you need to enable some basic security measures for the MariaDB database installation, by running the **mysql_secure_installation** script which ships with the **MariaDB** package.

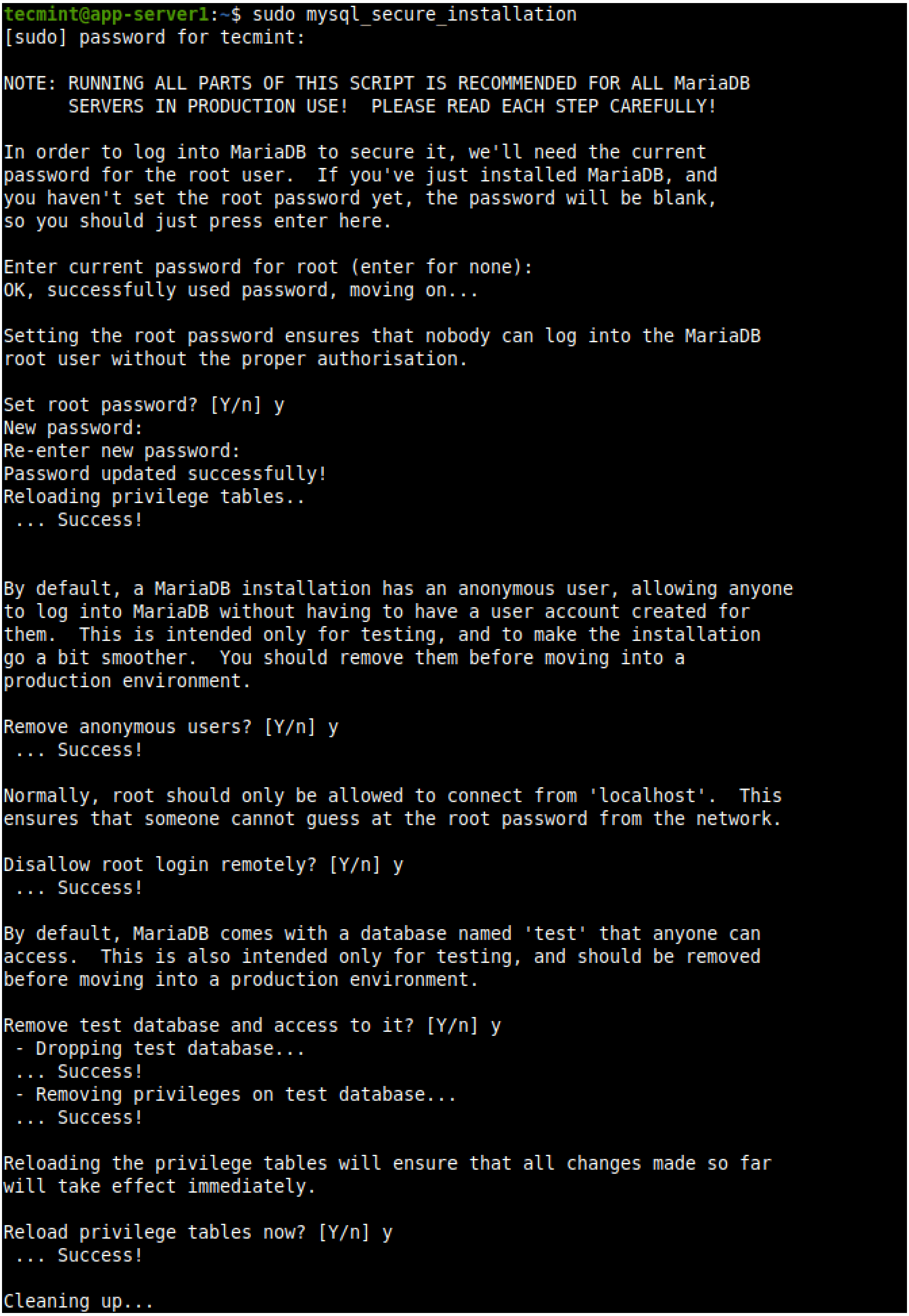
```
$ sudo mysql_secure_installation
```

After running the script, it will take you through a series of questions where you can answer yes(y) or no(n) to enable some security options. Because the database system has just been installed, there is no database root (or administrator) user password.

So you need to create one as shown in the following screenshot.

- Enter current password for root (enter for none): Enter
- Set a root password? [Y/n] y
- Remove anonymous users? [Y/n] y
- Disallow root login remotely? [Y/n] y
- Remove test database and access to it? [Y/n] y

- Reload privilege tables now? [Y/n] y

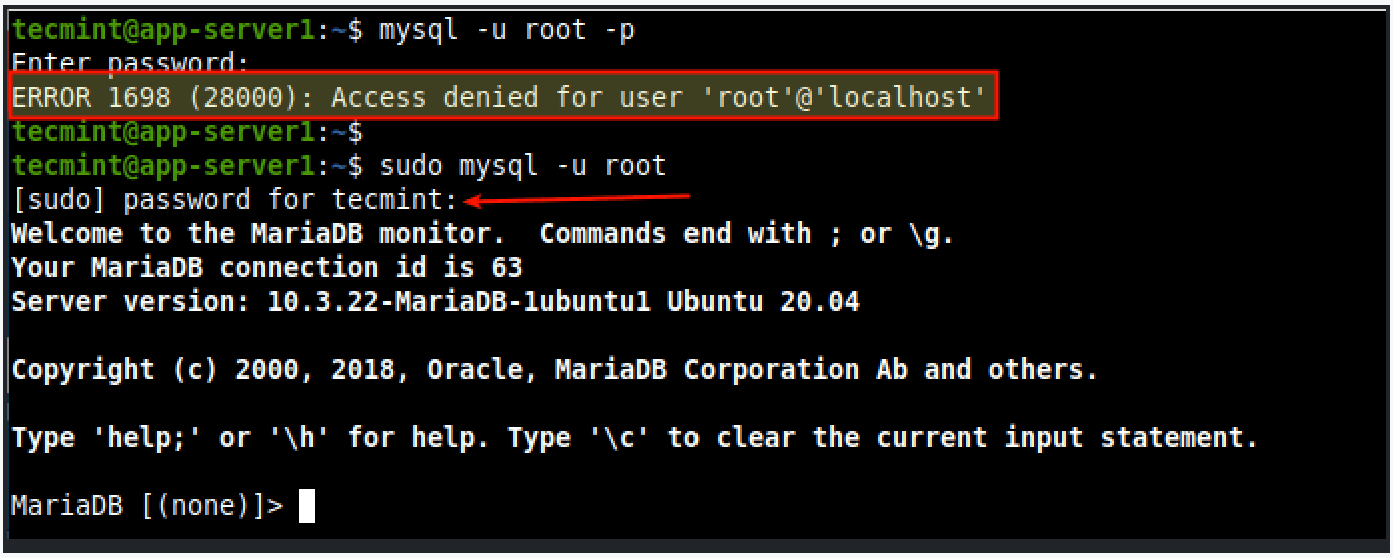


```
tecmint@app-server1:~$ sudo mysql_secure_installation
[sudo] password for tecmint:

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] y
 ... 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] y
 ... 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] y
 - Dropping test database...
 ... Success!
 - 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] y
 ... Success!

Cleaning up...
```

Secure MariaDB Server

8. To access the **MariaDB** shell, run the **mysql** command with the -u option with **sudo**. If you do not use the **sudo** command, you are bound to encounter the error indicated in the following screenshot.

```
$ mysql -u root -p
$ sudo mysql -u root
```



```
tecmint@app-server1:~$ mysql -u root -p
Enter password:
ERROR 1698 (28000): Access denied for user 'root'@'localhost'
tecmint@app-server1:~$
tecmint@app-server1:~$ sudo mysql -u root
[sudo] password for tecmint:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 63
Server version: 10.3.22-MariaDB-1ubuntu1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]>
```

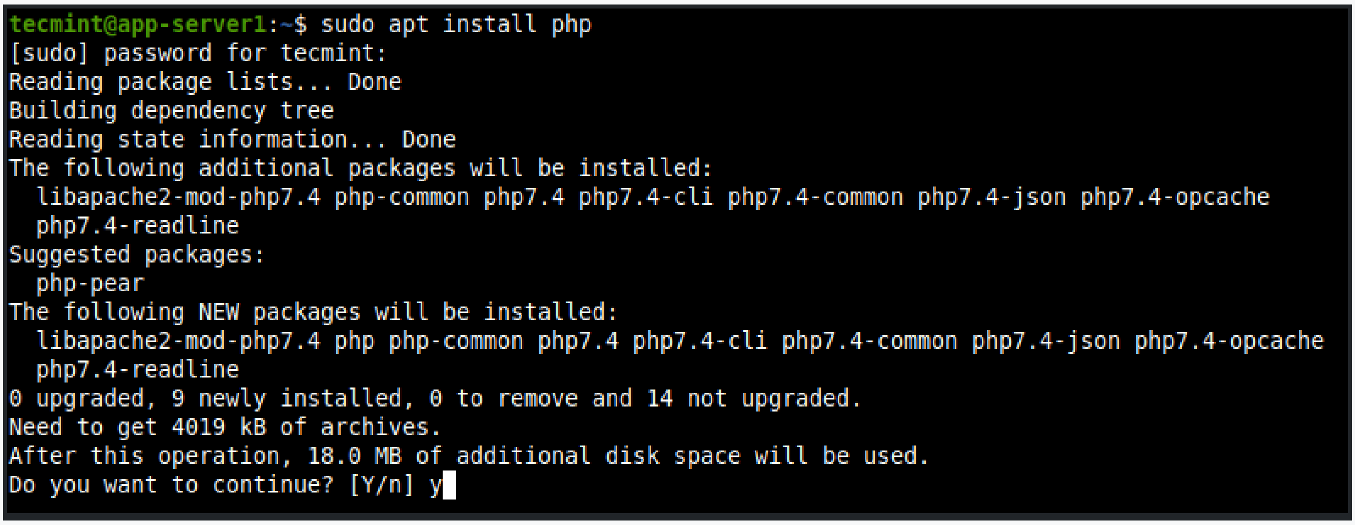
Access MariaDB Shell

Step 3: Installing PHP in Ubuntu 20.04

9. A general-purpose open-source scripting language, **PHP** is one of the most popular programming languages for web development. It powers some of the most popular websites and web applications in the world.

To install PHP, run the following command.

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



```
tecmint@app-server1:~$ sudo apt install php
[sudo] password for tecmint:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.4 php-common php7.4 php7.4-cli php7.4-common php7.4-json php7.4-opcache
  php7.4-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php7.4 php php-common php7.4 php7.4-cli php7.4-common php7.4-json php7.4-opcache
  php7.4-readline
0 upgraded, 9 newly installed, 0 to remove and 14 not upgraded.
Need to get 4019 kB of archives.
After this operation, 18.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Install PHP in Ubuntu 20.04

The PHP configuration file will be located in **/etc/php/7.2/**.

Also, depending on your project, you may want to install some PHP extensions required by your application. You can search a PHP extension as shown.

```
$ sudo apt-cache search php | grep php-                  #show all php packages
```

10. After finding the extension, you can install it. For example, I am installing PHP modules for Redis in-memory cache and Zip compression tool.

```
$ sudo apt install php-redis php-zip
```

11. After installing PHP extension, you need to restart apache to apply recent changes.

```
$ sudo systemctl restart apache2
```

12. Next, test if **Apache** is working in conjunction with **PHP**. Create an info.php page under the web document root **/var/www/html/** directory as shown.

```
$ sudo vi /var/www/html/info.php
```

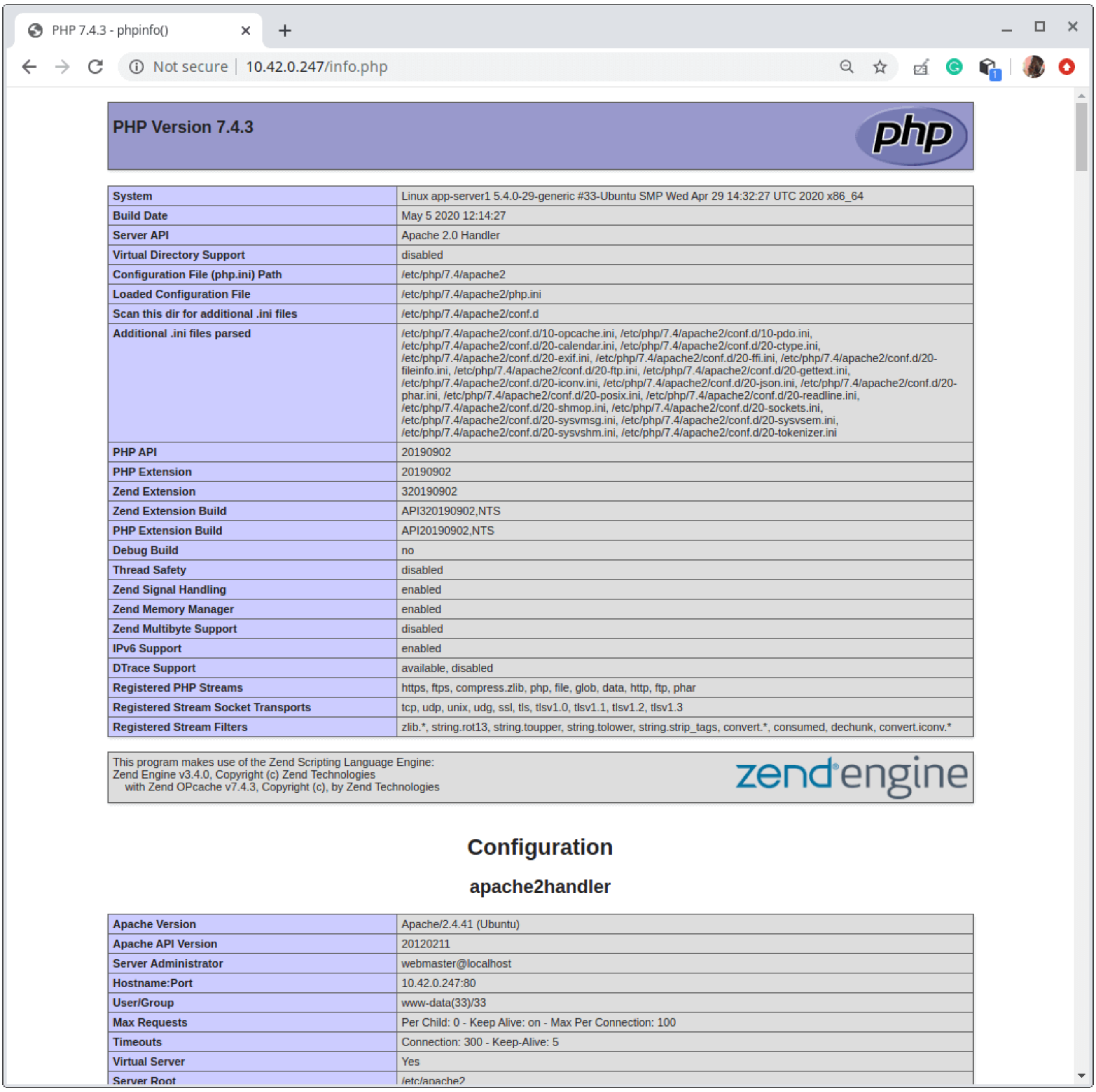
Copy and paste the following code in the file, then save the file and exit it.

```
<?php
        phpinfo();
?>
```

13. Next, open a web browser and navigate using the following address.

```
http://YOUR_SERVER_IP/info.php
```

If **Apache** and **PHP** are working well together, you should see the PHP information (configuration settings and available predefined variables, installed modules, and more on your system) shown in the following screenshot.



PHP Version 7.4.3

System	Linux app-server1 5.4.0-29-generic #33-Ubuntu SMP Wed Apr 29 14:32:27 UTC 2020 x86_64
Build Date	May 5 2020 12:14:27
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.4.0, Copyright (c) Zend Technologies
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies

Configuration

apache2handler

Apache Version	Apache/2.4.41 (Ubuntu)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	10.42.0.247:80
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2

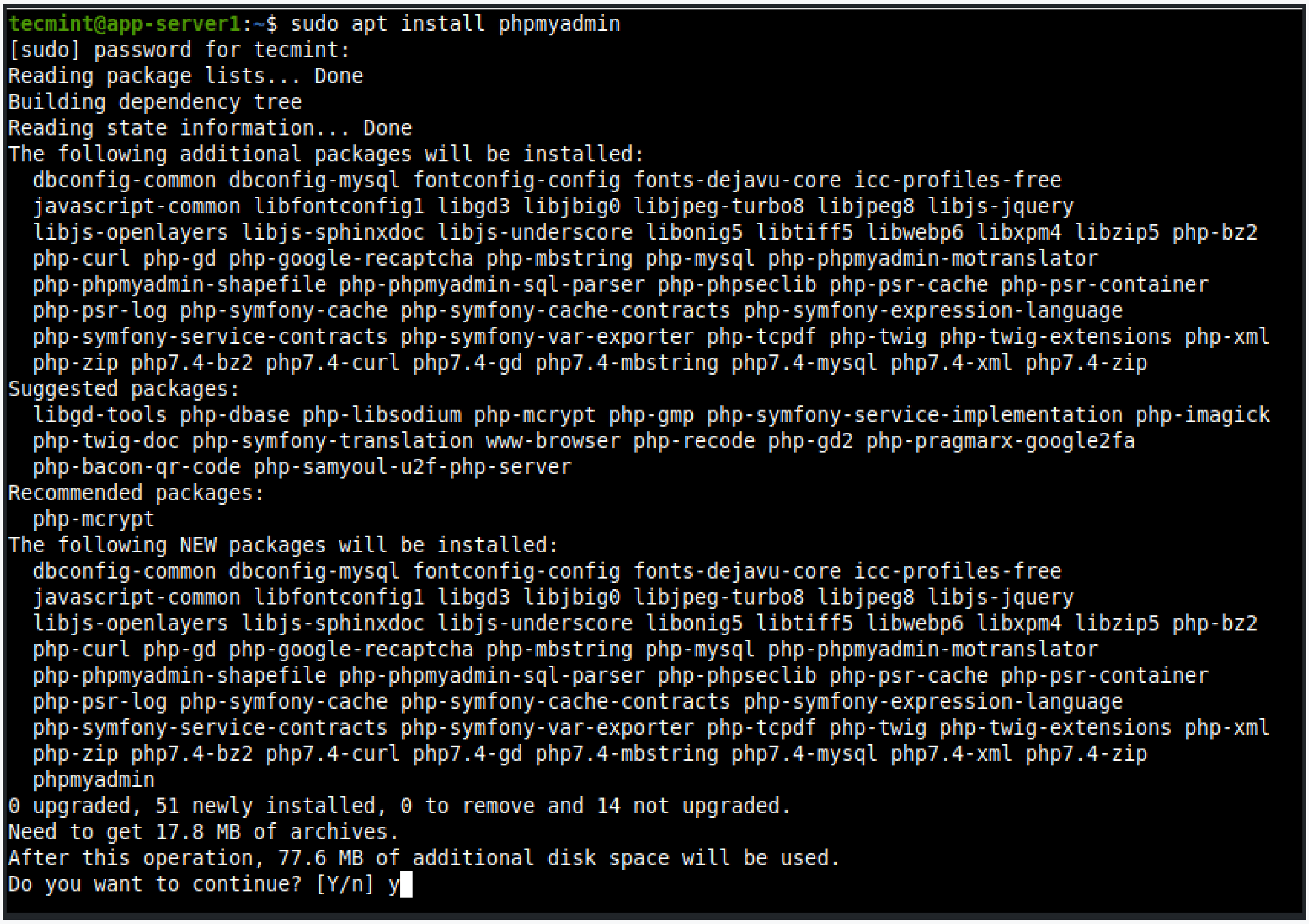
Verify PHP Information

Step 4: Installing PhpMyAdmin in Ubuntu 20.04

14. Intended to handle the administration of **MySQL/MariaDB** databases, **PhpMyAdmin** is a free widely-used web-based graphical tool with an intuitive web interface, that supports a wide range of operations on **MySQL** and **MariaDB**.

To install **PhpMyAdmin**, run the following command.

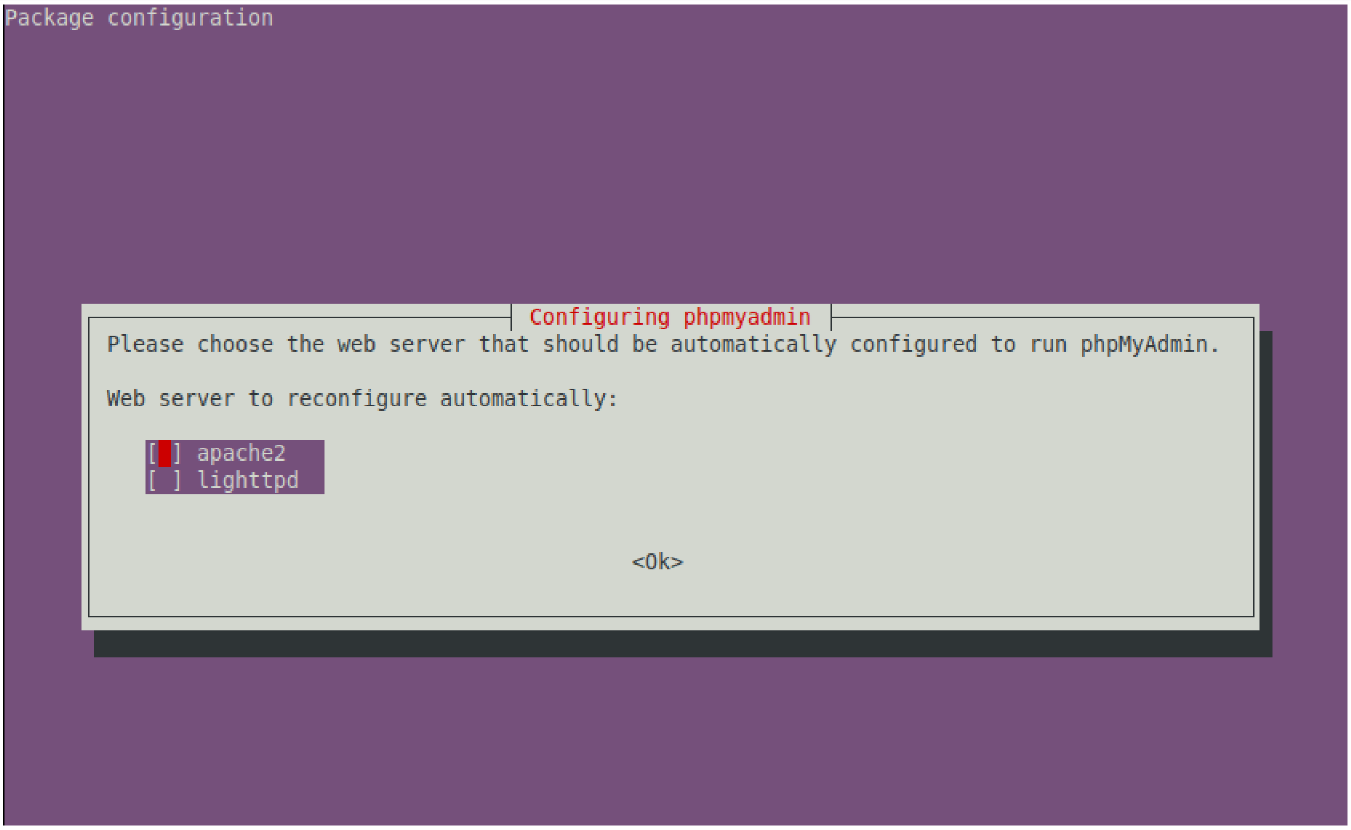
```
$ sudo apt install phpmyadmin
```



```
tecmint@app-server1:~$ sudo apt install phpmyadmin
[sudo] password for tecmint:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core icc-profiles-free
  javascript-common libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery
  libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5 libtiff5 libwebp6 libxpm4 libzip5 php-bz2
  php-curl php-gd php-google-recaptcha php-mbstring php-mysql php-phpmyadmin-motranslator
  php-phpmyadmin-shapefile php-phpmyadmin-sql-parser php-phpseclib php-psr-cache php-psr-container
  php-psr-log php-symfony-cache php-symfony-cache-contracts php-symfony-expression-language
  php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twig php-twig-extensions php-xml
  php-zip php7.4-bz2 php7.4-curl php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xml php7.4-zip
Suggested packages:
  libgd-tools php-dbase php-libsodium php-mcrypt php-gmp php-symfony-service-implementation php-imagick
  php-twig-doc php-symfony-translation www-browser php-recode php-gd2 php-pragmarx-google2fa
  php-bacon-qr-code php-samyoul-u2f-php-server
Recommended packages:
  php-mcrypt
The following NEW packages will be installed:
  dbconfig-common dbconfig-mysql fontconfig-config fonts-dejavu-core icc-profiles-free
  javascript-common libfontconfig1 libgd3 libjbig0 libjpeg-turbo8 libjpeg8 libjs-jquery
  libjs-openlayers libjs-sphinxdoc libjs-underscore libonig5 libtiff5 libwebp6 libxpm4 libzip5 php-bz2
  php-curl php-gd php-google-recaptcha php-mbstring php-mysql php-phpmyadmin-motranslator
  php-phpmyadmin-shapefile php-phpmyadmin-sql-parser php-phpseclib php-psr-cache php-psr-container
  php-psr-log php-symfony-cache php-symfony-cache-contracts php-symfony-expression-language
  php-symfony-service-contracts php-symfony-var-exporter php-tcpdf php-twig php-twig-extensions php-xml
  php-zip php7.4-bz2 php7.4-curl php7.4-gd php7.4-mbstring php7.4-mysql php7.4-xml php7.4-zip
  phpmyadmin
0 upgraded, 51 newly installed, 0 to remove and 14 not upgraded.
Need to get 17.8 MB of archives.
After this operation, 77.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Install PhpMyAdmin in Ubuntu 20.04

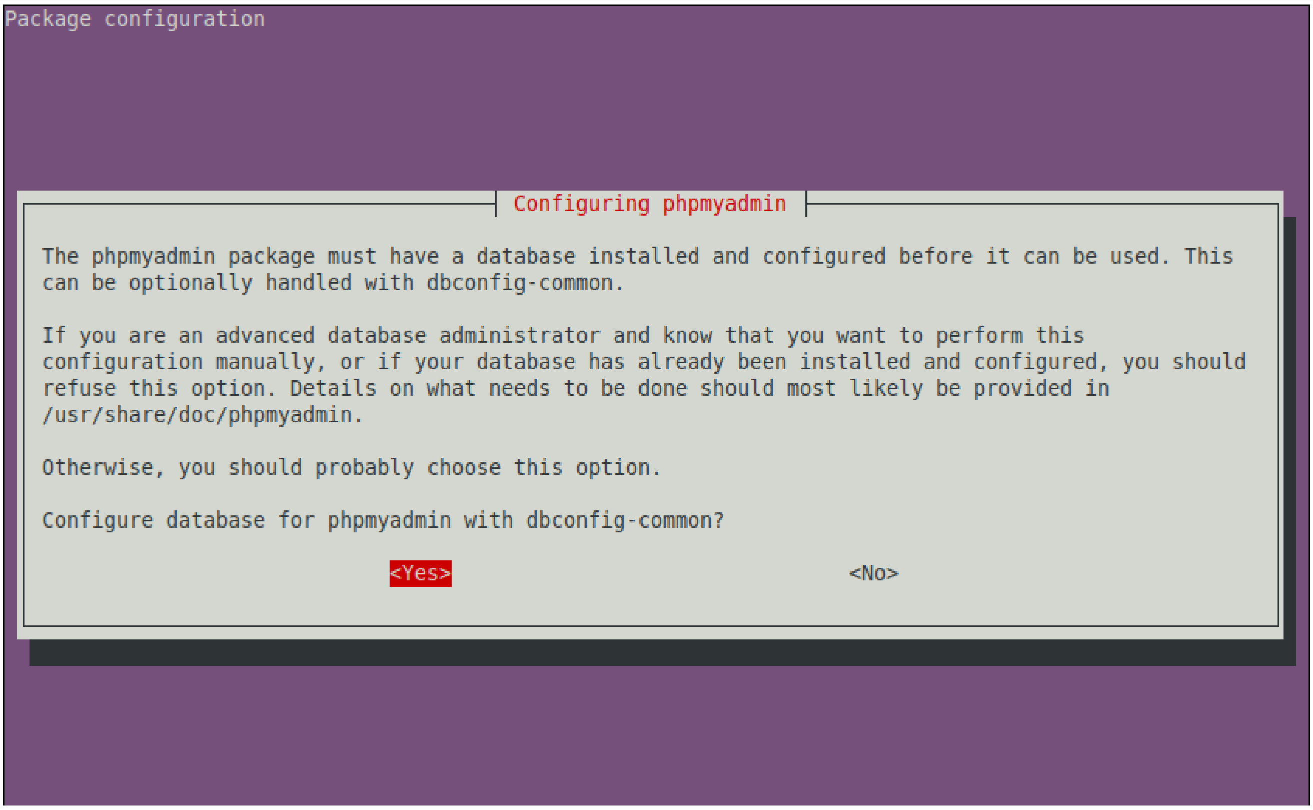
15. During the package installation, you will be prompted to choose the web server that should be automatically configured to run **PhpMyAdmin**. Click enter to use Apache, the default option.



```
Package configuration

                 ┤ Configuring phpmyadmin ├
 Please choose the web server that should be automatically configured to run phpMyAdmin.

 Web server to reconfigure automatically:

    [ ] apache2
    [ ] lighttpd

                               <Ok>
```

Configure Apache to Use PhpMyAdmin

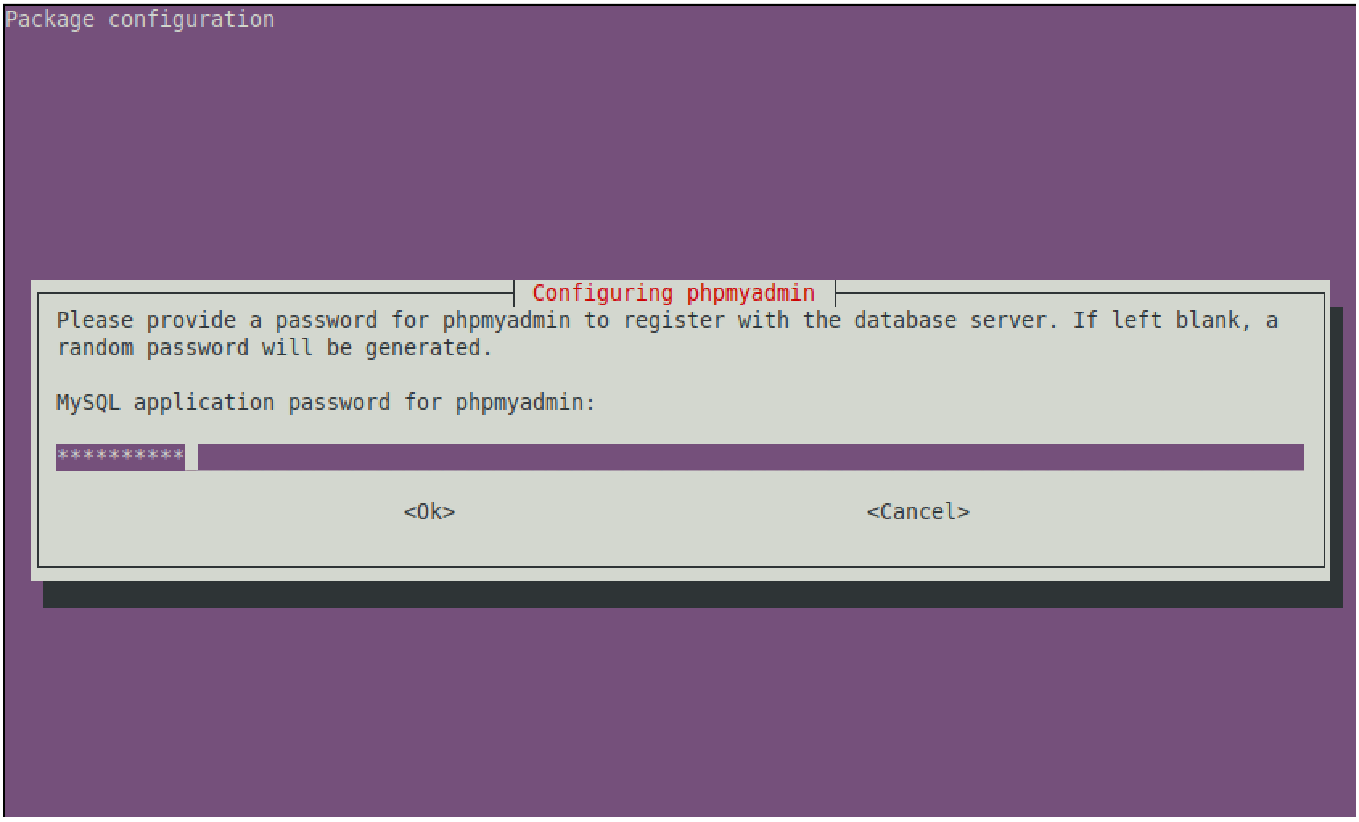
16. Also, **PhpMyAdmin** must have a database installed and configured before you can start using it. To configure a database for **PhpMyAdmin** with the **dbconfig-common** package, select **yes** in the next prompt.



```
Package configuration

                          ┤ Configuring phpmyadmin ├

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

 If you are an advanced database administrator and know that you want to perform this
 configuration manually, or if your database has already been installed and configured, you should
 refuse this option. Details on what needs to be done should most likely be provided in
 /usr/share/doc/phpmyadmin.

 Otherwise, you should probably choose this option.

 Configure database for phpmyadmin with dbconfig-common?

                      <Yes>                                   <No>
```

Configure Database for PhpMyAdmin

17. Next, create a password for **PhpMyAdmin** to register with the **MariaDB** database server.



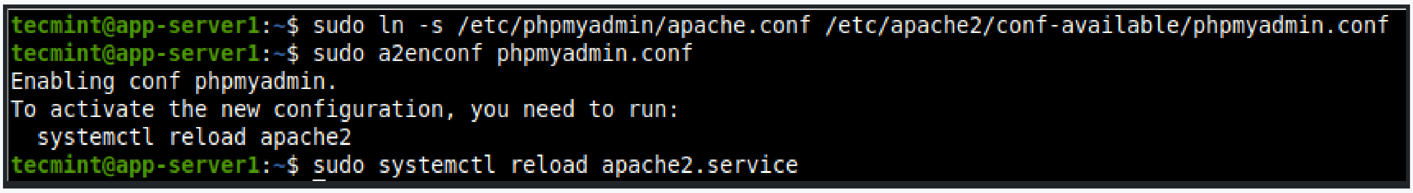
```
Package configuration

Configuring phpmyadmin
Please provide a password for phpmyadmin to register with the database server. If left blank, a
random password will be generated.

MySQL application password for phpmyadmin:

**********

<Ok>                              <Cancel>
```

Create a Password for PhpMyAdmin

Once the installation process is complete, the configuration files for phpMyAdmin are located in **/etc/phpmyadmin** and its main configuration file is **/etc/phpmyadmin/config.inc.php**. Another important configuration file is **/etc/phpmyadmin/apache.conf**, used to configure Apache2 to work with **PhpMyAdmin**.

18. Next, you need to configure **Apache2** to serve the **phpMyAdmin** site. Run the following command to symlink the file **/etc/phpmyadmin/apache.conf** to **/etc/apache2/conf-available/phpmyadmin.conf**. Then enable the **phpmyadmin.conf** configuration files for Apache2 and restart the **Apache2** service to apply the recent changes.

```
$ sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
$ sudo a2enconf phpmyadmin.conf
$ sudo systemctl reload apache2.service
```



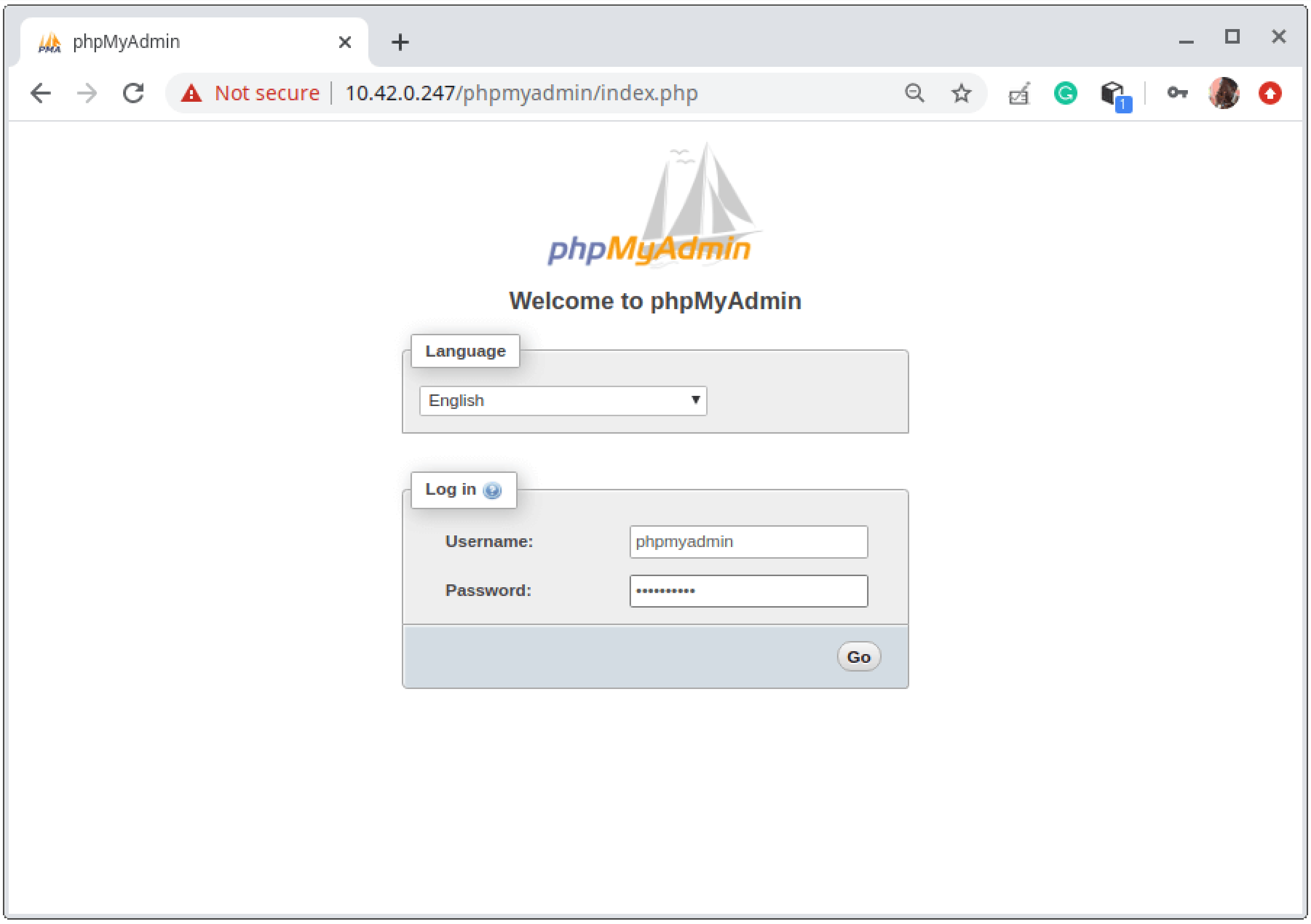
```
tecmint@app-server1:~$ sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf-available/phpmyadmin.conf
tecmint@app-server1:~$ sudo a2enconf phpmyadmin.conf
Enabling conf phpmyadmin.
To activate the new configuration, you need to run:
  systemctl reload apache2
tecmint@app-server1:~$ sudo systemctl reload apache2.service
```

Enable PhpMyAdmin for Apache2

19. In a browser go to **http://SERVER_IP/phpmyadmin**, replacing **SERVER_IP** with the server's actual IP address.

```
http://SERVER_IP/phpmyadmin
```

Once the **PhpMyAdmin** login page loads, enter **root** for the username and its password, or another MariaDB user, if you have any setup, and enter the user's password. If you disabled remote root user login, you can use the **phpmyadmin** user and password to log in.



Welcome to phpMyAdmin

Language

English

Log in

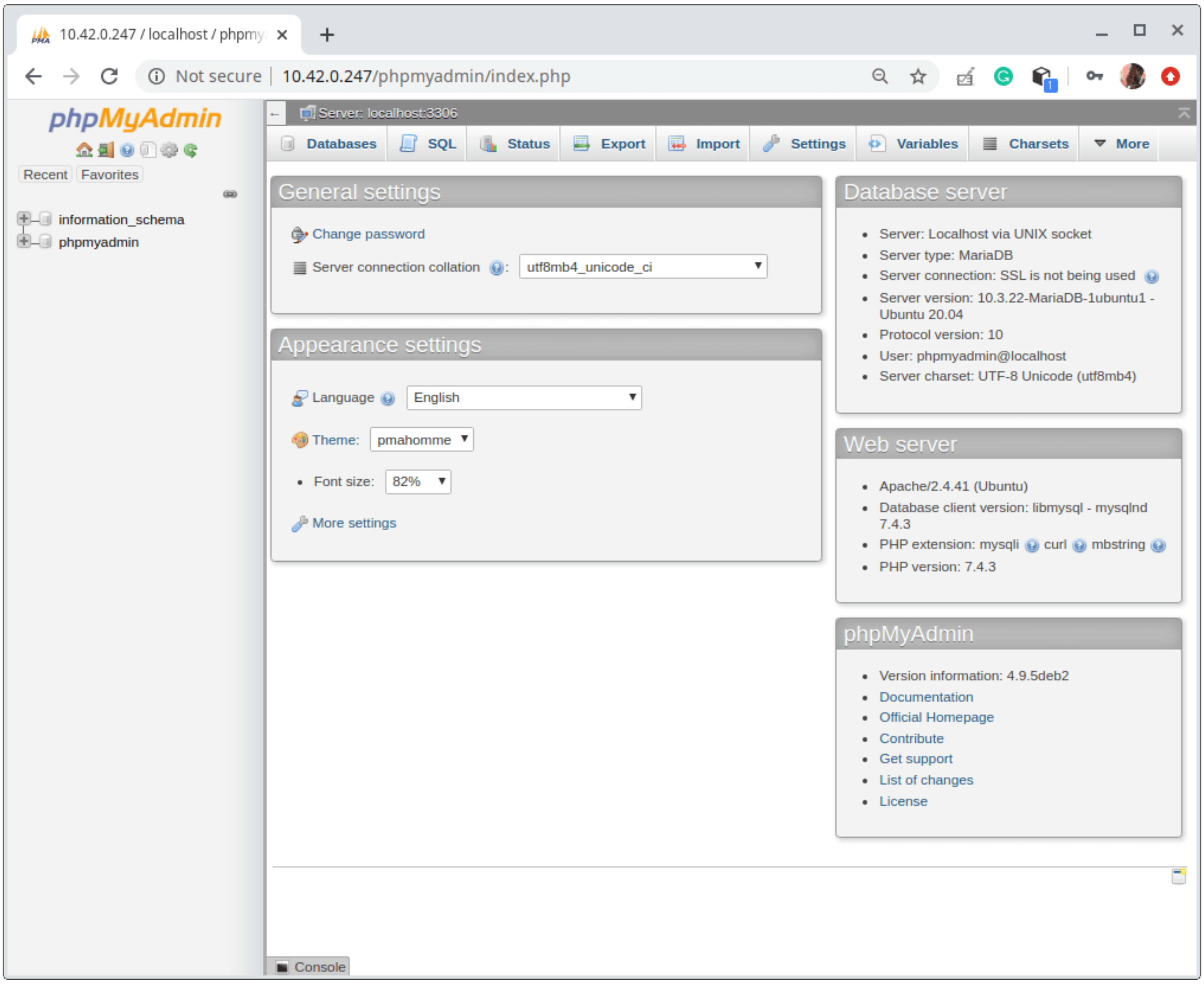
Username: phpmyadmin

Password: ••••••••••

Go

PhpMyAdmin Login

20. After login, you will see the **PhpMyAdmin** dashboard. Use it for managing databases, tables, columns, relations, indexes, users, permissions, etc.



General settings

Change password

Server connection collation: utf8mb4_unicode_ci

Appearance settings

Language: English

Theme: pmahomme

- Font size: 82%

More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MariaDB
- Server connection: SSL is not being used
- Server version: 10.3.22-MariaDB-1ubuntu1 - Ubuntu 20.04
- Protocol version: 10
- User: phpmyadmin@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.41 (Ubuntu)
- Database client version: libmysql - mysqlnd 7.4.3
- PHP extension: mysqli curl mbstring
- PHP version: 7.4.3

phpMyAdmin

- Version information: 4.9.5deb2
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

PhpMyAdmin Dashboard

This brings us to the end of this guide. Use the feedback form to ask any questions about this guide or any other **LAMP** stack related issues concerning **Ubuntu 20.04**.