Configure LAMP on CentOS 7 / RHEL 7

By **Raj** Last updated **Jan 16, 2016**

LAMP stands for Linux, Apache,MySQL and PHP, of course Linux can be any variant (Debian/Redhat), here is the small tutorial about installing the your own web server for the testing environment in latest release CentOS 7 / RHEL 7. In this latest CentOS, MySQL replaced by MariaDB as a default database.

Environment:

OS: Redhat Enterprise Linux 7 / CentOS 7

HostName: server.itzgeek.com

Ipaddress: 192.168.2.100

Install Linux:

Here is the article about Step by Step installation of CentOS 7 / RHEL 7. Now you have Linux, next is to install Apache,MySQL and PHP on it. Lets install one by one.

Install Apache:

To start off we will install Apache.

Step 1: Open up the Terminal and Switch to root user.

```
[raj@SRV01 ~]$ su
```

Step 2: Package name of the Apache is httpd. Install httpd using YUM. Type following Command in the terminal and then press enter.

```
[root@SRV01 ~]# yum install httpd
```

Step 3: Start the Apache by using the following command.

```
[root@SRV01 ~]# systemctl start httpd.service
```

Step 4: To make the apache to start during the every boot, Type the following on terminal and hit Enter.

```
[root@SRV01 ~]# systemctl enable httpd.service
```

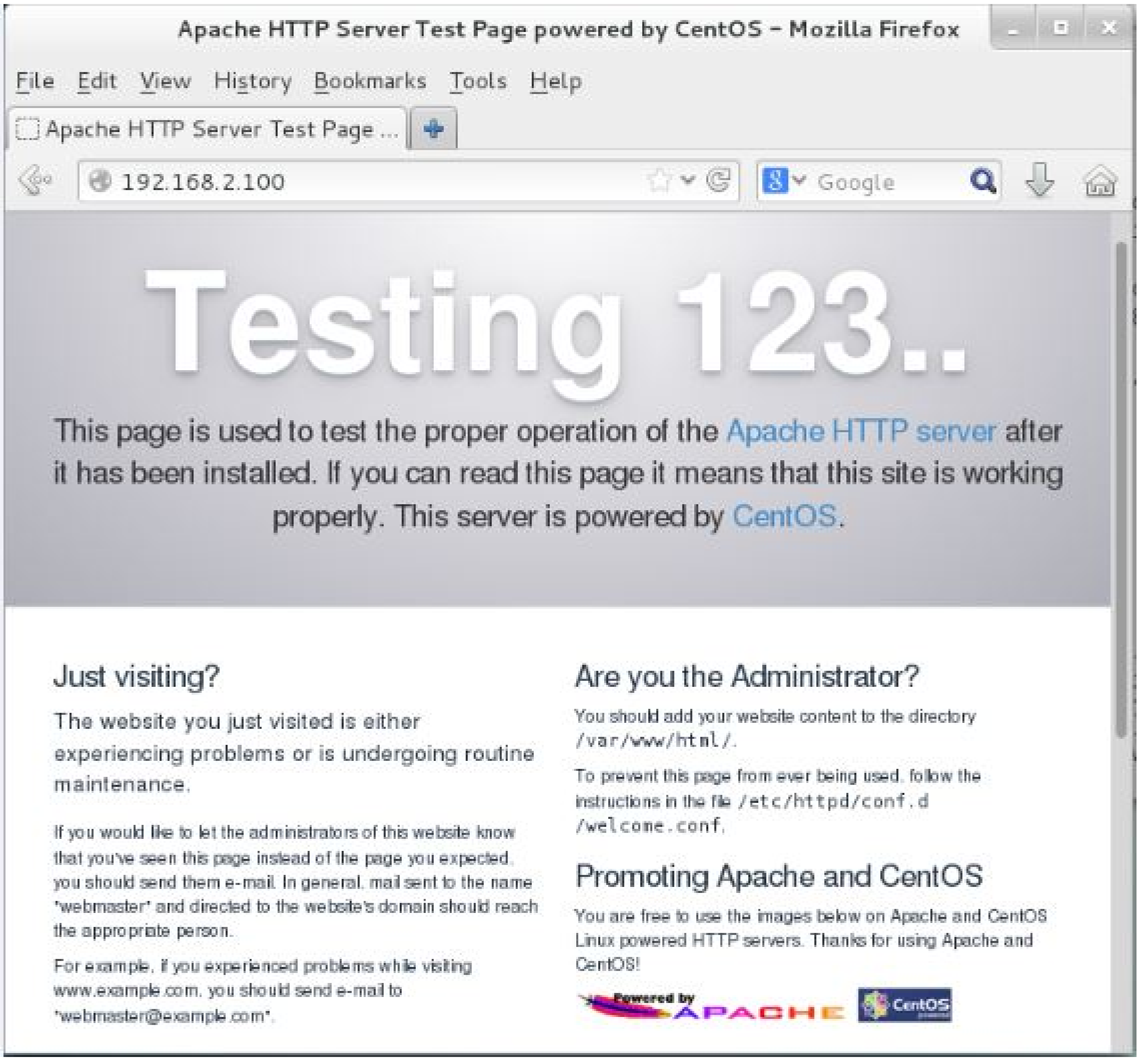
Testing Apache:

To make sure everything installed correctly we will now test Apache to ensure it is working properly.

1. Open up any web browser and then enter the following into the web address:

```
 http://localhost/  or  http://192.168.2.100
```

You will get the web page saying “Testing 123”, this confirms that the Apache is working fine. Apache’s default document root is /var/www/html on CentOS, the configuration file is /etc/httpd/conf/httpd.conf. Additional configurations are stored in the /etc/httpd/conf.d/ directory.


Apache HTTP Server Test Page powered by CentOS - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Apache HTTP Server Test Page ...
192.168.2.100
Google
Powered by APACHE
CentOS

Testing 123..

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page it means that this site is working properly. This server is powered by CentOS.

Just visiting?

The website you just visited is either experiencing problems or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

Are you the Administrator?

You should add your website content to the directory /var/www/html/.

To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.

Promoting Apache and CentOS

You are free to use the images below on Apache and CentOS Linux powered HTTP servers. Thanks for using Apache and CentOS!

CentOS 7 – Apache Default Page

Install MySQL:

Next is to install the MySQL on the Linux, now it is a MariaDB package.

Step 1: Open the Terminal.

Step 2: Type the following command and then Press Enter.

```
[root@SRV01 ~]#  yum install mariadb mariadb-server
```

Start MySQL server.

```
[root@SRV01 ~]# systemctl start mariadb.service
```

Step 3: To make the MySQL to start during the every boot, Type the following on terminal and hit Enter.

```
[root@SRV01 ~]# systemctl enable mariadb.service
```

Nex is to make the MariaDB secure by using the mysql_secure_installation command.

This program enables you to improve the security of your MariaDB installation in the following ways:

- You can set a password for `root` accounts.
- You can remove `root` accounts that are accessible from outside the local host.
- You can remove anonymous-user accounts.
- You can remove the `test` database (which by default can be accessed by all users, even anonymous users), and privileges that permit anyone to access databases with names that start with `test_`.

```
[root@server1 ~]# 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):  <— Enter Current root
password
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   <— Set root password
New password:   <— Enter root password
Re-enter new password:     <— Re enter root 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    <— Remove anonymous user
… 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    <— root remote login
… 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   <— Remove the test database
- 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    <— Save the changes
… 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!
```

Install PHP:

By default Apache server supports the HTML language only not PHP, for that we need to install PHP. To install PHP please follow the steps.

Step 1: Type following line into Terminal and press enter: This command includes support package for the MySQL.

```
[root@SRV01 ~]# yum install php php-mysql
```

Step 2. You need to restart the server after the installation of the PHP, to do that type the following on the terminal.

```
[root@SRV01 ~]# systemctl restart httpd.service
```

Testing PHP:

To test PHP, place a PHP file on to the default directory of the Apache. The document root of the default web site is /var/www/html. We will now create a small PHP file (info.php) in that directory and call it in a browser. The file will display lots of useful details about our PHP installation, such as the installed PHP version.

Step 1. In the terminal copy/paste the following line:

```
[root@SRV01 ~]# vi /var/www/html/info.php
```

This will open up a file called *info.php*.

Step 2. Copy/Paste this line into the phpinfo file:

```
<?php
```

```
phpinfo();
```

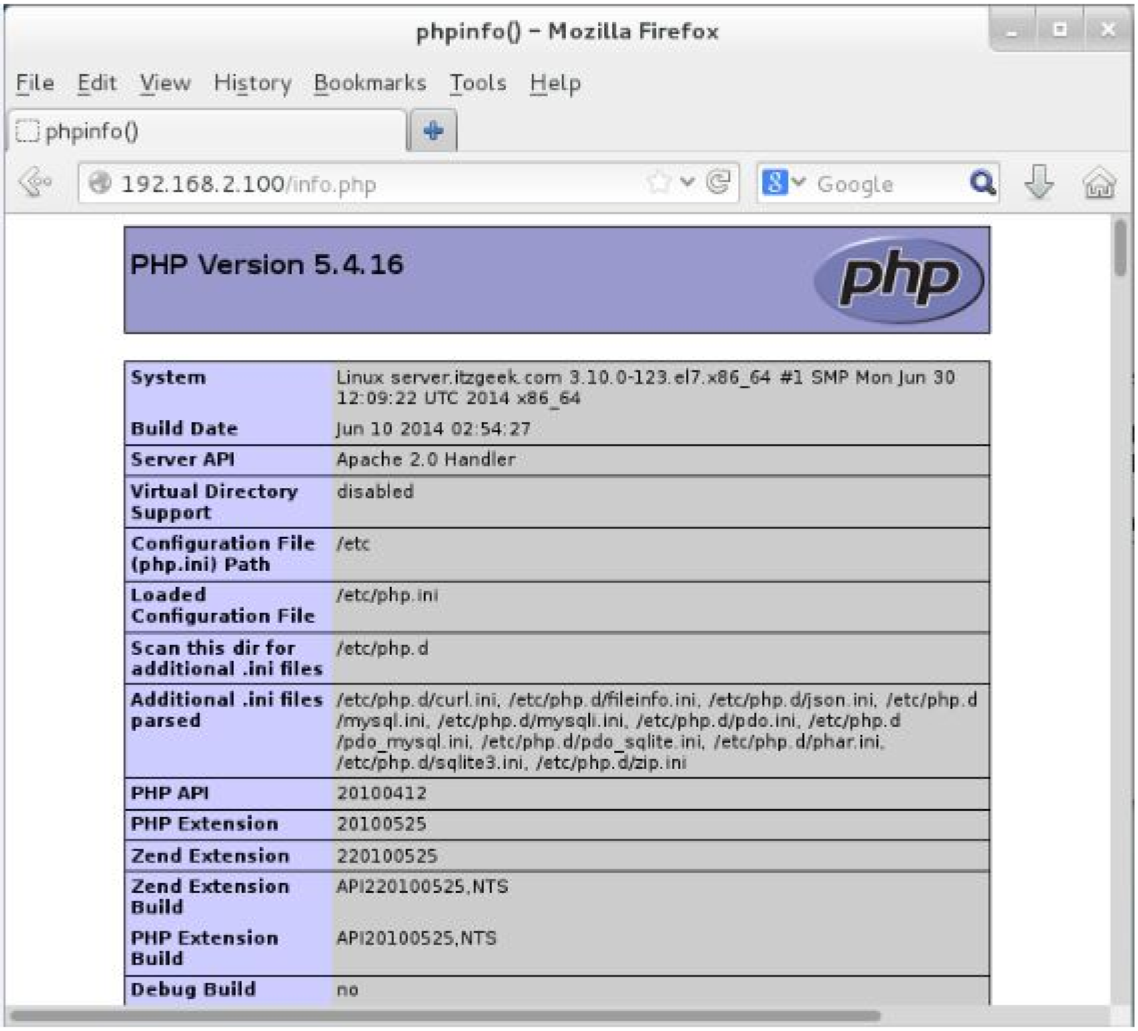
```
?>
```

Step 3. Save and close the file. use **Esc + :wq** for saving the file.

Step 4. Now open you’re web browser and type the following into the web address:

```
http://localhost/info.php or http://192.168.2.100/info.php
```

The page look like below:

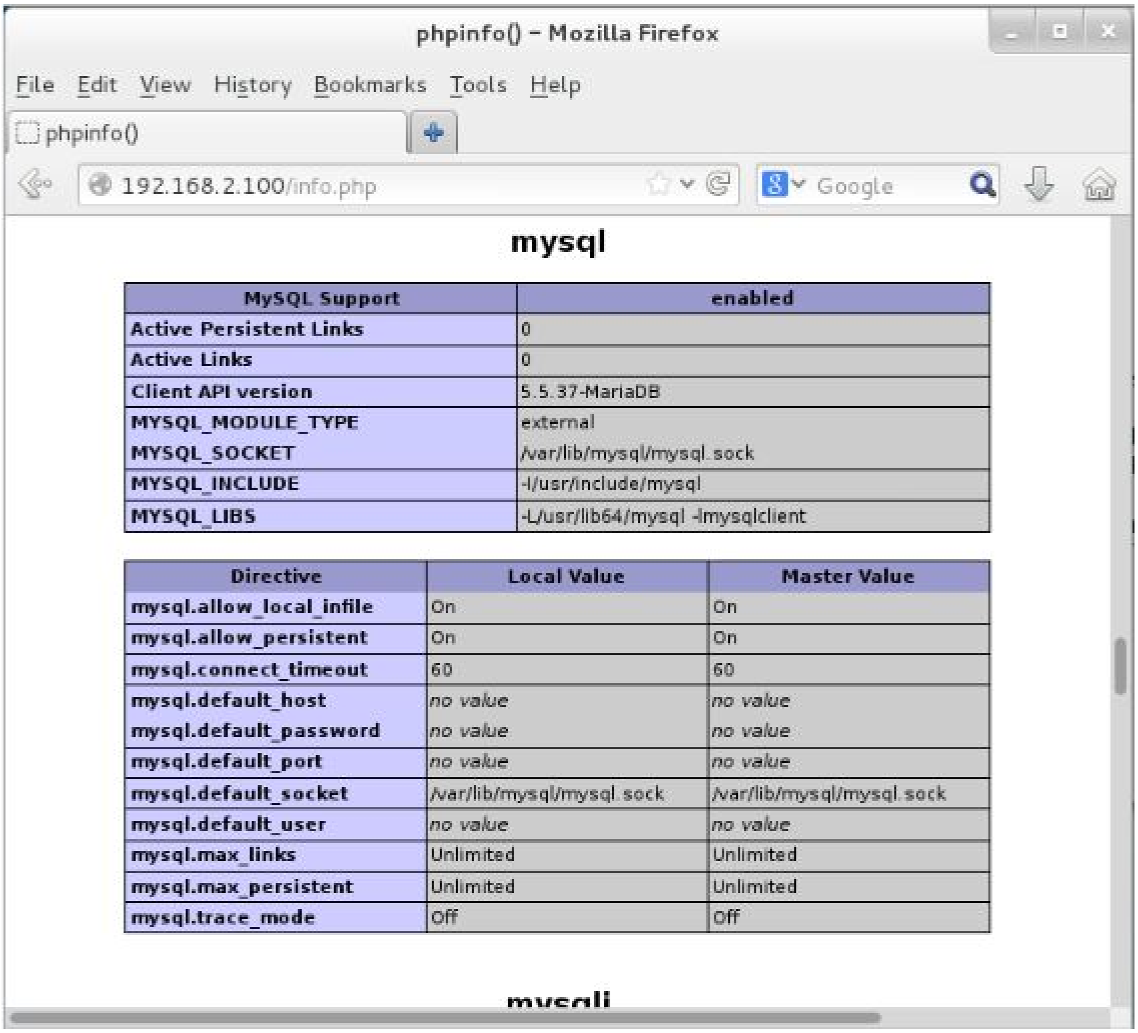

phpinfo() - Mozilla Firefox
File Edit View History Bookmarks Tools Help
phpinfo()
192.168.2.100/info.php
Google
php

PHP Version 5.4.16

System	Linux server.itzgeek.com 3.10.0-123.el7.x86_64 #1 SMP Mon Jun 30 12:09:22 UTC 2014 x86_64
Build Date	Jun 10 2014 02:54:27
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525,NTS
PHP Extension Build	API20100525,NTS
Debug Build	no

CentOS 7 – PHPInfo Server Details

Scroll down the browser to modules section to check the support for the MariaDB. you will get the screen like below.


phpinfo() - Mozilla Firefox
File Edit View History Bookmarks Tools Help
phpinfo()
192.168.2.100/info.php
Google
mysqli

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	5.5.37-MariaDB
MYSQL_MODULE_TYPE	external
MYSQL_SOCKET	/var/lib/mysql/mysql.sock
MYSQL_INCLUDE	-I/usr/include/mysql
MYSQL_LIBS	-L/usr/lib64/mysql -lmysqlclient

Directive	Local Value	Master Value
mysql.allow_local_infile	On	On
mysql.allow_persistent	On	On
mysql.connect_timeout	60	60
mysql.default_host	*no value*	*no value*
mysql.default_password	*no value*	*no value*
mysql.default_port	*no value*	*no value*
mysql.default_socket	/var/lib/mysql/mysql.sock	/var/lib/mysql/mysql.sock
mysql.default_user	*no value*	*no value*
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited
mysql.trace_mode	Off	Off

CentOS 7 – PHPInfo MariaDB Details

That's All!, you have successfully installed the LAMP on CentOS 7 / RHEL 7.

apache centos 7 mariadb mysql php rhel 7 web server

3 Comments **ITzGeek** 1 **Login**

♡ **Recommend** **Share** **Sort by Newest**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Jaybond • 3 years ago

this is not about configure. This is installation

• Reply • Share ›

Cody ➦ Jaybond • 3 years ago

Yes, I agree. I used this tutorial to setup LAMP on my CentOS 7 based VPS and while it is working perfectly fine I found the following similar article https://www.rosehosting.com... which I was able to easily adapt and use to further configure my LAMP setup.

I really think that this article should include some more configuration of LAMP like vhosts, security hardening, disabling not needed modules etc...

but anyway though, pretty solid article there. thanks.

• Reply • Share ›

Jem • 4 years ago

Many Thanks. Excellent, precise tutorial. Got me out of a bind. Thanks again!

• Reply • Share ›

ALSO ON **ITZGEEK**

How to Upgrade To Ubuntu 18.04 From Ubuntu 16.04 / Ubuntu 17.10 [Detailed

1 comment • a month ago

Avatar **chrisdbarnett** — No No No No No!!!! Do not do this!!!! There is a fat chance it won't boot if you've come up from 16.04 LTS or 17.10

OpenStack Pike – Single Node OpenStack Installation on CentOS 7 / RHEL 7

1 comment • 5 months ago

Avatar **DEEPAK** — I was searching for this tutorial only ... for the past one week.. i mean for pike installation...Finally I got it.Thank you very

add-apt-repository command not found – Debian & Ubuntu – Quick Fix

1 comment • 9 months ago

Avatar **Moises Hernandez Hernandez** — thanks!

How to Install Webmin on Debian 9 / Ubuntu 16.04

3 comments • a year ago

Avatar **ITzGeek Web** — Thank you. Same has been updated.

Designed By: Publisher. Once Powered By: Kinsta. Delivered By: Key CDN