How To Install Linux, Apache, MySQL, PHP (LAMP) stack On CentOS 6

*Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]*
Saved From: http://faq.asphosthelpdesk.com/article.php?id=171

LAMP stack is a group of open source software used to get web servers up and running. The acronym stands for Linux, Apache, MySQL, and PHP. Since the server is already running CentOS, the linux part is taken care of. Here is how to install the rest.

Set Up

The steps in this tutorial require the user on the virtual private server to have root privileges.

Step One—Install Apache

Apache is a free open source software which runs over 50% of the world"s web servers.

To install apache, open terminal and type in this command:

```
sudo yum install httpd
```

Once it installs, you can start apache running on your server:

```
sudo service httpd start
```

That"s it. To check if Apache is installed, direct your browser to your server"s IP address (eg. http://12.34.56.789). The page should display the words "It works!" like [this](#).

How to find your Server"s IP address

You can run the following command to reveal your server"s IP address.

```
ifconfig eth0 | grep inet | awk '{ print ifconfig eth0 | grep inet | awk '{ print $2 }' }'
```

Step Two—Install MySQL

MySQL is a powerful database management system used for organizing and retrieving data on a virtual server

To install MySQL, open terminal and type in these commands:

```
sudo yum install mysql-server
sudo service mysqld start
```

During the installation, MySQL will ask you for your permission twice. After you say Yes to both, MySQL will install.

Once it is done installing, you can set a root MySQL password:

```
sudo /usr/bin/mysql_secure_installation
```

The prompt will ask you for your current root password.

Since you just installed MySQL, you most likely won"t have one, so leave it blank by pressing enter.

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

Then the prompt will ask you if you want to set a root password. Go ahead and choose Y and follow the instructions.

CentOS automates the process of setting up MySQL, asking you a series of yes or no questions.

It"s easiest just to say Yes to all the options. At the end, MySQL will reload and implement the new changes.

```
By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL 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, MySQL 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...
```

```
All done!  If you've completed all of the above steps, your MySQL
installation should now be secure.
Thanks for using MySQL!
```

Step Three—Install PHP

PHP is an open source web scripting language that is widely used to build dynamic webpages.

To install PHP on your virtual private server, open terminal and type in this command:

```
sudo yum install php php-mysql
```

Once you answer yes to the PHP prompt, PHP will be installed.

PHP Modules

PHP also has a variety of useful libraries and modules that you can add onto your server. You can see the libraries that are available by typing:

```
yum search php-
```

Terminal then will display the list of possible modules. The beginning looks like this:

```
php-bcmath.x86_64 : A module for PHP applications for using the bcmath library
php-cli.x86_64 : Command-line interface for PHP
php-common.x86_64 : Common files for PHP
php-dba.x86_64 : A database abstraction layer module for PHP applications
php-devel.x86_64 : Files needed for building PHP extensions
php-embedded.x86_64 : PHP library for embedding in applications
php-enchant.x86_64 : Human Language and Character Encoding Support
php-gd.x86_64 : A module for PHP applications for using the gd graphics library
php-imap.x86_64 : A module for PHP applications that use IMAP
```

To see more details about what each module does, type the following command into terminal, replacing the name of the module with whatever library you want to learn about.

```
yum info  name of the module
```

Once you decide to install the module, type:

```
sudo yum install  name of the module
```

You can install multiple libraries at once by separating the name of each module with a space.

Congratulations! You now have LAMP stack!

We should also set the processes to run automatically when the server boots (php will run automatically once Apache starts):

```
sudo chkconfig httpd on
sudo chkconfig mysqld on
```

Step Four—RESULTS: See PHP on your Server

Although LAMP is installed on your virtual server, we can still take a look and see the components online by creating a quick php info page

To set this up, first create a new file:

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

Add in the following line:

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

Then Save and Exit.

Restart apache so that all of the changes take effect on your virtual server:

```
sudo service httpd restart
```

Finish up by visiting your php info page (make sure you replace the example ip address with your correct one): http://12.34.56.789/info.php