How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu

Authored by: **ASPHostServer Administrator** *[asphostserver@gmail.com]*
Saved From: http://faq.asphosthelpdesk.com/article.php?id=242

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 virtual private server is already running Ubuntu, the linux part is taken care of. Here is how to install the rest.

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 these commands:
```
sudo apt-get update
sudo apt-get install apache2
```
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

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

```
sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql
```
During the installation, MySQL will ask you to set a root password. If you miss the chance to set the password while the program is installing, it is very easy to set the password later from within the MySQL shell.

Once you have installed MySQL, we should activate it with this command:
```
sudo mysql_install_db
```
Finish up by running the MySQL set up script:
```
sudo /usr/bin/mysql_secure_installation
```
The prompt will ask you for your current root password.

Type it in.

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

Then the prompt will ask you if you want to change the root password. Go ahead and choose N and move on to the next steps.

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...
```

Once you're done with that you can finish up by installing PHP.

Step Three—Install PHP PHP is an open source web scripting language that is widely use to build dynamic webpages.

To install PHP, open terminal and type in this command.
```
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```
After you answer yes to the prompt twice, PHP will install itself.

It may also be useful to add php to the directory index, to serve the relevant php index files:
```
sudo nano /etc/apache2/mods-enabled/dir.conf
```
Add index.php to the beginning of index files. The page should now look like this:
```
<IfModule mod_dir.c>
        DirectoryIndex index.php index.html index.cgi index.pl index.php
index.xhtml index.htm
</IfModule>
```
PHP Modules PHP also has a variety of useful libraries and modules that you can add onto your virtual server. You can see the libraries that are available.
```
apt-cache search php5-
```
Terminal will then display the list of possible modules. The beginning looks like this:
```
php5-cgi - server-side, HTML-embedded scripting language (CGI binary)
php5-cli - command-line interpreter for the php5 scripting language
```

```
php5-common - Common files for packages built from the php5 source
php5-curl - CURL module for php5
php5-dbg - Debug symbols for PHP5
php5-dev - Files for PHP5 module development
php5-gd - GD module for php5
php5-gmp - GMP module for php5
php5-ldap - LDAP module for php5
php5-mysql - MySQL module for php5
php5-odbc - ODBC module for php5
php5-pgsql - PostgreSQL module for php5
php5-pspell - pspell module for php5
php5-recode - recode module for php5
php5-snmp - SNMP module for php5
php5-sqlite - SQLite module for php5
php5-tidy - tidy module for php5
php5-xmlrpc - XML-RPC module for php5
php5-xsl - XSL module for php5
php5-adodb - Extension optimising the ADOdb database abstraction library
php5-auth-pam - A PHP5 extension for PAM authentication
[...]
```
Once you decide to install the module, type:
```
sudo apt-get 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!

 Step Four—RESULTS: See PHP on your Server *Although LAMP is installed, 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/info.php
```
 Add in the following line:
```
<?php
phpinfo();
?>
```
 Then Save and Exit.

 Restart apache so that all of the changes take effect:
```
sudo service apache2 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