

How To Install Linux, Nginx, MySQL, PHP (LEMP) Stack on Debian 7

Authored by: **ASPHostServer Administrator** [asphostserver@gmail.com]

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=194>

Lemp Stack: Basic Info

LEMP stack is a group of open source software to get a server up and running. The acronym includes the Linux (L), MySQL (M), and PHP (P) of the more traditional LAMP stack, but switches out Apache for Nginx (pronounced engine x).

This tutorial explains how to install nginx/MySQL/PHP-fpm, as well as all of the required repos to download nginx and php-fpm. It will also review how to configure php and nginx, as well as how to create a php info page.

1) Update Apt-Get

The apt-get update command is used to re-synchronize the package index files from their sources. If used in combination with the apt-get upgrade command, they install the newest versions of all packages currently available.

At the moment, we only need to do a thorough update:

```
sudo apt-get update
```

2) Install MySQL on your server

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 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...

The prompt will ask if you want to change the root password. Go ahead and choose N for this option, as the root password should already be set; however, for the rest of the questions you can simply reply Y to all-- unless there is a reason for you to do otherwise.

Now let's install nginx!

3) Install and Configure Nginx on your server

Installation

Initial installation is simple with the apt-get command.

```
sudo apt-get install nginx
```

nginx needs a command to begin running:

```
sudo service nginx start
```

Now if you point your browser to your IP address, it should confirm that nginx was successfully installed on your cloud server.

*Run the following command to reveal your server's IP address.

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

Configuration

Open up the default virtual host file with this command:

```
sudo nano /etc/nginx/sites-available/default
```

The configuration should include the changes below (the details of the changes are under the config information):

UPDATE: Newer Ubuntu versions create a directory called 'html' instead of 'www' by default. If /usr/share/nginx/www does not exist, it's probably called html. Make sure you update your configuration appropriately.

```
[...]
server {
    listen    80;

    root /usr/share/nginx/www;
```

```

index index.php index.html index.htm;
server_name example.com;
location / {
    try_files $uri $uri/ /index.html;
}
error_page 404 /404.html;
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root /usr/share/nginx/www;
}
# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
location ~ \.php$ {
    try_files $uri =404;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    include fastcgi_params;
}
}
[...]
```

Here are the details of the changes:

Add index.php to the index line.

Change the server_name from local host to your domain name or IP address
(replace the example.com in the configuration)

Change the correct lines in "location ~ \.php\$ {" section

Save and Exit

4) Install and Configure PHP

Installation

You probably guessed it! We will use the **apt-get** command to install PHP-FPM:

```
sudo apt-get install php5-fpm
```

Configuration

We need to make one small change in the php configuration. Open up php.ini:

```
sudo nano /etc/php5/fpm/php.ini
```

Find the line

cgi.fix_pathinfo=1 and **change the 1 to 0.**

```
cgi.fix_pathinfo=0
```

If this number is kept as 1, the php interpreter will do its best to process the file that is as near to the requested file as possible. This is a possible security risk. If this number is set to 0, conversely, the interpreter will only process the exact file path— a much safer alternative. Save and Exit.

We need to make another small change in the php5-fpm configuration. Open up www.conf:

```
sudo nano /etc/php5/fpm/pool.d/www.conf
```

Find the line, listen = 127.0.0.1:9000, and change the 127.0.0.1:9000 to /var/run/php5-fpm.sock.

```
listen = /var/run/php5-fpm.sock
```

Save and Exit.

Restart php-fpm:

```
sudo service php5-fpm restart
```

5) Create a Php Info Page

We can quickly see all of the details of the new php configuration.

To set this up, first create a new file:

```
sudo nano /usr/share/nginx/www/info.php
```

Add in the following line:

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

Then Save and Exit.

Finishing Up

Restart nginx:

```
sudo service nginx restart
```

You can see the nginx and php-fpm configuration details by visiting <http://youripaddress/info.php>

Your LEMP stack is now set up and configured on your virtual private server :)