

# How To Install (LEMP) nginx, MySQL, PHP stack on Arch Linux

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

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=218>

---

## About Lemp

LEMP stack is a group of open source software to get web servers up and running. The acronym stands for Linux, nginx (pronounced Engine x), MySQL, and PHP. Since the server is already running Arch Linux, the linux part is taken care of. Here is how to install the rest.

## Step One—Pacman

Because pacman, the arch package manager, has a rolling package release, we should update Arch and its repositories before proceeding with any other steps:

```
sudo pacman -Syu
```

## Step Two—Install MySQL

Once everything is fresh and up to date, we can start to install the server software, beginning with MySQL.

```
sudo pacman -S mysql
```

Once MySQL installs, start both mysql and the secure installation process. You will also be able to set the MySQL root password during the installation.

```
sudo systemctl start mysqld && mysql_secure_installation
```

When initially prompted for the MySQL root password, you can go ahead and press enter, as it has not yet been set. Your installation should look like this:

```
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 MySQL
root user without the proper authorization.
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

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!
```

Follow up by restarting MySQL:

```
sudo systemctl restart mysqld
```

### **Step Three—Install nginx**

Once MySQL is all set up, we can move on to installing nginx on the server.

```
sudo pacman -S nginx
```

nginx does not start on its own. To get nginx running, type:

```
sudo systemctl start nginx
```

You can confirm that nginx has installed as your web server by directing your browser to your IP address. You can run the following command to reveal your server's IP address.

```
curl -s icanhazip.com
```

### **Step Four—Install PHP-FPM**

In order to process php application, we will need to install php-fpm.

```
sudo pacman -S php-fpm
```

Once its installed, start it up.

```
sudo systemctl start php-fpm
```

Finally, we need to tell nginx to run php using php-fpm. To accomplish this, first open up the nginx configuration file:

```
sudo nano /etc/nginx/nginx.conf
```

Find the location block that deals with php applications and replace the text in the section with the following:

```
location ~ /\.php$ {
    fastcgi_pass    unix:/var/run/php-fpm/php-fpm.sock;
    fastcgi_index  index.php;
    root           /srv/http;
    include        fastcgi.conf;
}
```

Save, exit, and restart nginx:

```
sudo systemctl restart nginx
```

## Step Five—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 /srv/http/info.php
```

Add in the following line:

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

Then Save and Exit.

Restart nginx

```
sudo systemctl restart nginx
```

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.

## Step Six—Configure the Daemons to Start at Boot

To ensure that all of the LEMP programs start automatically after any server restarts:

```
sudo systemctl enable nginx mysqld php-fpm
```

With that, LEMP is installed.