About LAMP

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 Fedora, the linux part is taken care of. Here is how to install the rest.

Setup

Before you start installing the LAMP programs, you should first download and install all of the updates with yum update:

```
yum update
```

Step One—Install Apache

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

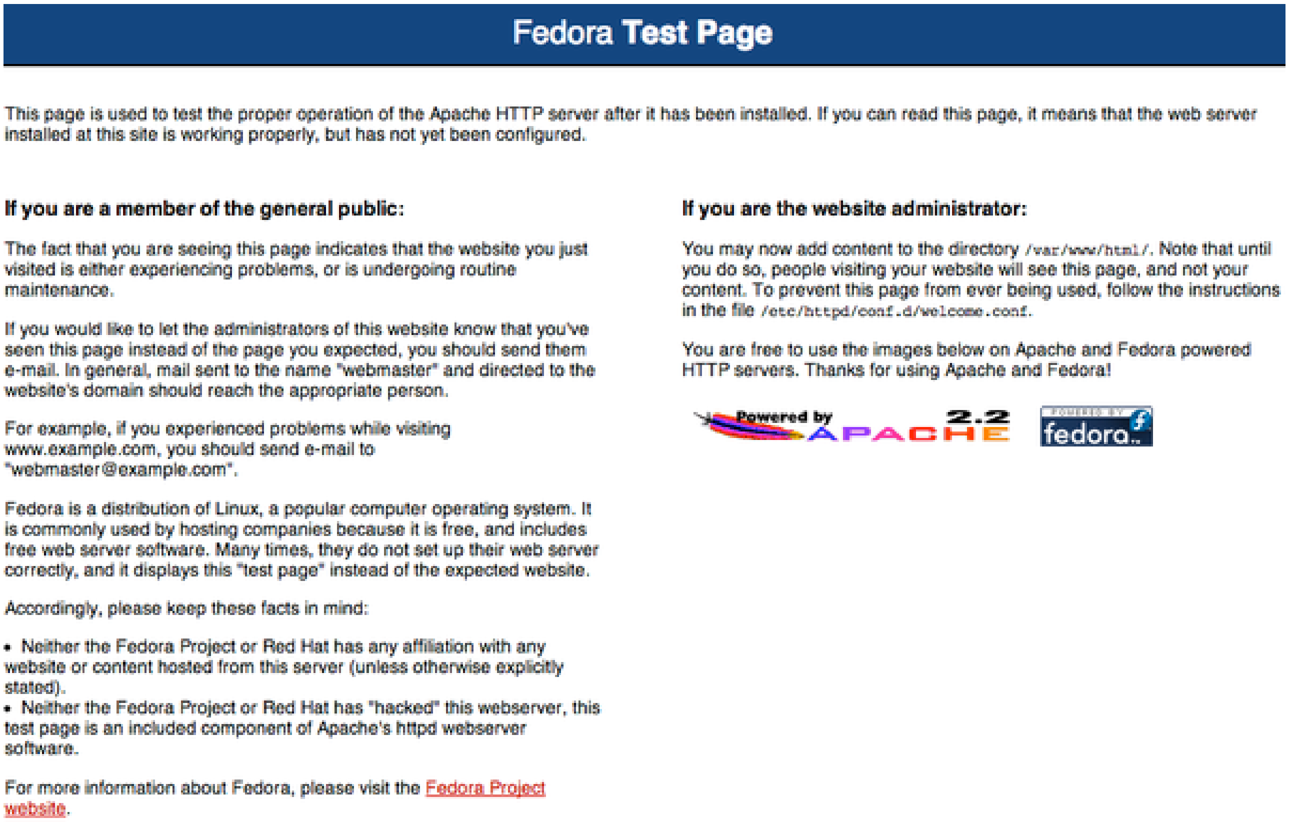
To install apache, open terminal and type in this command:

```
sudo yum install httpd
```

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

```
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). You should see the default Fedora page



Fedora Test Page

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 the web server installed at this site is working properly, but has not yet been configured.

If you are a member of the general public:

The fact that you are seeing this page indicates that 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".

Fedora is a distribution of Linux, a popular computer operating system. It is commonly used by hosting companies because it is free, and includes free web server software. Many times, they do not set up their web server correctly, and it displays this "test page" instead of the expected website.

Accordingly, please keep these facts in mind:

- Neither the Fedora Project or Red Hat has any affiliation with any website or content hosted from this server (unless otherwise explicitly stated).
- Neither the Fedora Project or Red Hat has "hacked" this webserver, this test page is an included component of Apache's httpd webserver software.

For more information about Fedora, please visit the Fedora Project website.

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

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

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 $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 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.

Fedora 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 install itself.

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-fpdf-doc.noarch : Documentation for php-fpdf
php-libvirt-doc.noarch : Document of php-libvirt
php-pear-Auth-radius.noarch : RADIUS support for php-pear-Auth
```

```
php-pear-Auth-samba.noarch : Samba support for php-pear-Auth
ice-php-devel.i686 : PHP tools for developping Ice applications
ice-php-devel.x86_64 : PHP tools for developping Ice applications
perl-PHP-Serialization.noarch : Converts between PHP's serialize() output and
                              : the equivalent Perl structure
php-IDNA_Convert.noarch : Provides conversion of internationalized strings to
                        : UTF8
php-Kohana.noarch : The Swift PHP Framework
php-LightweightPicasaAPI.noarch : A lightweight API for Picasa in PHP
php-PHPMailer.noarch : PHP email transport class with a lot of features
php-Smarty.noarch : Template/Presentation Framework for PHP
php-ZendFramework.noarch : Leading open-source PHP framework
php-ZendFramework-Auth-Adapter-Ldap.noarch : Zend Framework LDAP
                                           : Authentication Adapter
php-ZendFramework-Cache-Backend-Apc.noarch : Zend Framework APC cache backend
```

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 on your droplet!

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();
?>
```

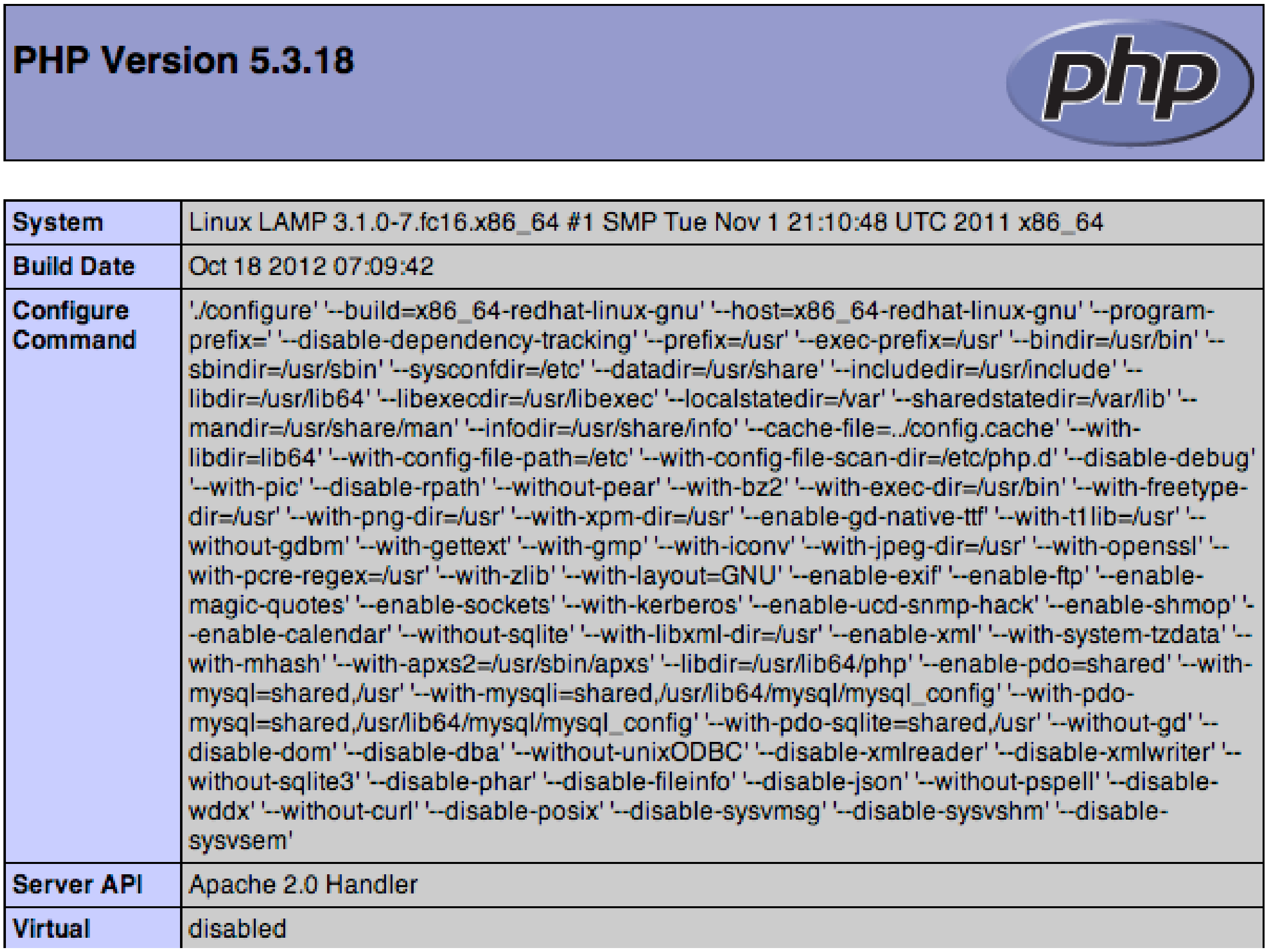
Then Save and Exit.

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

It should look similar to this:



PHP Version 5.3.18

System	Linux LAMP 3.1.0-7.fc16.x86_64 #1 SMP Tue Nov 1 21:10:48 UTC 2011 x86_64
Build Date	Oct 18 2012 07:09:42
Configure Command	'./configure' '--build=x86_64-redhat-linux-gnu' '--host=x86_64-redhat-linux-gnu' '--program-prefix=' '--disable-dependency-tracking' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib64' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/var/lib' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--cache-file=../config.cache' '--with-libdir=lib64' '--with-config-file-path=/etc' '--with-config-file-scan-dir=/etc/php.d' '--disable-debug' '--with-pic' '--disable-rpath' '--without-pear' '--with-bz2' '--with-exec-dir=/usr/bin' '--with-freetype-dir=/usr' '--with-png-dir=/usr' '--with-xpm-dir=/usr' '--enable-gd-native-ttf' '--with-t1lib=/usr' '--without-gdbm' '--with-gettext' '--with-gmp' '--with-iconv' '--with-jpeg-dir=/usr' '--with-openssl' '--with-pcre-regex=/usr' '--with-zlib' '--with-layout=GNU' '--enable-exif' '--enable-ftp' '--enable-magic-quotes' '--enable-sockets' '--with-kerberos' '--enable-ucd-snmp-hack' '--enable-shmop' '--enable-calendar' '--without-sqlite' '--with-libxml-dir=/usr' '--enable-xml' '--with-system-tzdata' '--with-mhash' '--with-apxs2=/usr/sbin/apxs' '--libdir=/usr/lib64/php' '--enable-pdo=shared' '--with-mysql=shared,/usr' '--with-mysqli=shared,/usr/lib64/mysql/mysql_config' '--with-pdo-mysql=shared,/usr/lib64/mysql/mysql_config' '--with-pdo-sqlite=shared,/usr' '--without-gd' '--disable-dom' '--disable-dba' '--without-unixODBC' '--disable-xmlreader' '--disable-xmlwriter' '--without-sqlite3' '--disable-phar' '--disable-fileinfo' '--disable-json' '--without-pspell' '--disable-wddx' '--without-curl' '--disable-posix' '--disable-sysvmsg' '--disable-sysvshm' '--disable-sysvsem'
Server API	Apache 2.0 Handler
Virtual	disabled