

How To Install Opigno on Debian 7 with Git and Drush

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

Saved From: <http://faq.asphosthelpdesk.com/article.php?id=288>

Introduction

Opigno is a new, fast growing e-learning platform based on the battle-hardened Drupal framework. Setting up Opigno via sFTP is pretty easy, but as you now have your own server, we should set it up using GIT and Drush.

Set up your server

It's recommended to create a new user that is inside the sudo group. This will allow us to administer our server without using our root user. If you don't know how to do this.

The listed steps are for users that are not *root* but are inside the sudo group. If you want to use your *root* user however, that's fine. Just don't type the `sudo` keyword in front of the commands listed in this tutorial.

Now, log in to you server:

```
ssh -l [your user] [your server ip]
```

Make sure your server is up to date

We are going to update our system and make sure it runs on the latest releases:

```
sudo apt-get update
sudo apt-get upgrade
```

Done. Now, we're going to set up our technology stack to run Opigno.

Install Apache

Opigno runs on NginX and Apache, but we'll use Apache in this tutorial. Use the following command to install it:

```
sudo apt-get install apache2
```

To check if Apache installed correctly, use your browser to navigate to your virtual server's IP address (e.g. <http://12.345.6.789>). You should see "It works !".

Next, we want to make sure Apache accepts per-directory configuration overrides with `.htaccess` files. Opigno comes with a default `.htaccess` file that configures the directory it's in. This will set some nice security

defaults.

Call the following:

```
sudo pico /etc/apache2/sites-available/default
```

Locate the group<Directory /var/www/>(not<Directory />).

Change theAllowOverride NonetoAllowOverride All. Hit Ctrl+x to quit, and type "y" to save.

Now we must restart Apache:

```
sudo service apache2 restart
```

Enable clean URLs

Opigno can use *clean URLs*, which turns a URL like `?=path/to/page` to `/path/to/page`.

This is fully optional, but still recommended. For this, we need to enable the Apache Rewrite module. Call the following:

```
sudo updatedb
sudo a2enmod rewrite
sudo service apache2 restart
```

Install PHP

Opigno runs on Drupal, which in turn runs on PHP. This will set up PHP, as well as the PHP GD library Drupal depends on.

```
sudo apt-get install php5 php5-gd
```

It is also recommended to install APC. APC will make your Opigno install run much faster, as it caches the PHP opcode in memory. Being a scripting language, PHP normally has to rebuild the opcode for every single request. This will make the opcode be built on the first request only:

```
sudo apt-get install php-apc
```

Check you PHP setup

If you want to make sure your PHP got installed correctly and linked with Apache, execute the following commands:

```
echo "<?php phpinfo();" > phpinfo.php
sudo mv phpinfo.php /var/www/phpinfo.php
```

Now, if you redirect your browser to `[your server ip]/phpinfo.php`, you will see your current PHP

setup.

After you checked that PHP works correctly, **remove the phpinfo.php file**, as it can expose sensitive information for hackers to exploit:

```
sudo rm /var/www/phpinfo.php
```

Install MySQL

Opigno can run on a different number of databases, but the most recommended one is MySQL, as many Drupal modules are only tested against MySQL-compatible databases. Thus, if you want to extend Opigno with third-party modules, you will be certain it works with your setup.

```
sudo apt-get install mysql-server php5-mysql
```

Next, we should clean up MySQL a bit to make it ready for production.

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

Create a new database user for Opigno

It is *highly recommended* to create a new MySQL user that will *only* have access to the Opigno database. This will increase security, especially if you're going to run other systems on your server as well. This is optional however, you *can* just use your root user...

First, log in to MySQL:

```
mysql -u root -p
```

You are now inside the MySQL command prompt. We will create a database for opigno.

```
create database opigno_db;
```

Now, we will create a new user and give it access to this database only.

```
grant all privileges on opigno_db.* to 'opigno'@'localhost' identified by  
'[enter a password]';
```

Now log out of the MySQL prompt:

```
exit
```

Install Drush

Drush stands for "Drupal Shell" and is a great way to manage any systems running on Drupal. Once you start using it, you will get addicted. To install Drush, run the following command:

```
sudo apt-get install drush
```

If you want to read more about Drush, you can find all the information you need on drush.ws.

Install git

Git is great for Opigno as it will help you to keep your site up to date easily and efficiently. It should come shipped with your Debian 7.0 install, but just to be safe, call:

```
sudo apt-get install git
```

Stack Ready!

Now you have all the packages to get started. We will now install Opigno. You will see it's easy.

Getting Opigno

We will change directory somewhere to keep the GIT repo nice and tidy. This will allow us to spawn multiple Opigno installs if needed (like testing and production). In this tutorial, we'll keep the cloned repo in a Projects folder in our users home folder (also written~), but you can organize your code however you see fit.

```
cd ~
mkdir Projects
cd Projects
```

At the time of writing, the latest branch of Opigno is [7.x-1.x](#). So we'll clone that one.

```
git clone --branch 7.x-1.x http://git.drupal.org/project/opigno_lms.git
cd opigno_lms
```

This will clone the latest and greatest version of Opigno. However, for production sites, you may want to use a specific release instead of the very latest code, as it might contain some unstable code. To see all releases, just type the following:

```
git tag -l
```

Pick one you want to use (e.g.: 1.0), and type:

```
git checkout 7.x-1.0
```

This will use the 1.0 version.

Building Opigno

Opigno is what is called a Drupal *distribution*. This means it aggregates many different modules and

installs/configures everything for you. This is very efficient, as all third-party modules are managed in separate GIT repositories. To fetch all the code needed to run Opigno, we need to *buildit*. That's where Drush comes in.

In this tutorial, we won't mess with Apache virtual hosts, as you might not have domain names ready. Ideally though, you might want to set up a virtual host for a test site and one for the production site (e.g.: my-opigno.com and dev.my-opigno.com). This would mean you use different web roots. But here we'll just use the Apache /var/www folder to contain our code.

Call the following:

```
cd /var/www
sudo drush make ~/Projects/opigno_lms/build-opigno-lms.make
```

Drush will ask you if you want to build Opigno in the current folder. Just say yes.

Finally, because you used `sudo`, all files will be assigned to root:root. We want it to be assigned to the user PHP will use, so call:

```
sudo chown www-data:www-data . -R
```

Install Opigno

You now have 2 options:

1. You can navigate to [your server ip] and use the graphical installer.
2. You can use Drush to install your site via the command line. We'll detail this second step.

If you moved out of the /var/www folder, move back into it:

```
cd /var/www
```

Now, call:

```
sudo drush site-install opigno_lms --db-url=mysql://[root or the new MySQL user
you created]:[password you provided]@localhost/[database name]
--account-pass=[your Opigno admin password]
```

Important note: if you did not enable the Apache Rewrite module as explained above, add this at the end of the `drush site-install` command:

```
--clean-url=0
```

Choose yes when prompted. The install will take a few seconds.

The install will give a warning about the PDF library used. You can safely ignore this.

Call this again to assign all files to the correct group:

```
sudo chown www-data:www-data . -R
```

Login

Now you can navigate your browser to `http://[your server ip]`. You will see a login form on the left. If you installed through drush, your username is *admin* and the password is the one you provided in the `--account-pass` parameter. If you used the graphical installer, you have provided your login information on the site configuration step.

"Access denied"

Depending on the version you checked out from GIT, you may see an "Access denied" message with a user login form to the left. This is completely normal, as you have to be logged in to access any content on Opigno. Just log in and start enjoying your new platform.

Set up a cron

Opigno can use a cron task to regularly cleanup stuff. This step is optional, as Opigno contains a fallback system that will do this automatically, but not as efficiently. For this reason, it's better to set up a cron task.

The cron task is protected by a unique key for security reasons. To find this key, navigate your browser to `[your server ip]/admin/reports/status` (if you're not logged in, you will be asked to).

Locate the line that says "Cron maintenance tasks" and copy the URL provided.

Now, from your command line, call:

```
sudo crontab -e
```

Add the following line, which will run the cron script every hour:

```
0 * * * * wget -O - -q -t 1 http://[your server ip]/cron.php?cron_key=[very-long-key]
```

Hit Ctrl+x to quit, type 'y' to save.

Restart the cron daemon:

```
sudo service cron restart
```

Rejoice !

Congratulations, you finished the tutorial ! You can now start using the Opigno platform! You can find more information on Opigno at opigno.org.