Installing Drupal v8/9 Website on Windows IIS localhost Server

Mike

Image not found or type unknown

by moa Saturday, July 17, 2021 - 21:51 comments
Coding images
Credits

Photo by Jeremy Bishop on Unsplash

For newcomers to Drupal CMS the first few questions to consider would often include

 Is Drupal the right CMS for you? Do you have the pre-requisites to

navigate its deployment? What learning is required to achieve optimum performance of the website?

For a simple blog site that does not require multiple users input and access control is not an issue, other [CMS](#) such as OpenCMS, WordPress, Joomla, Typo3 etc could easily meet these requirements. CMS' could be cloud-based or installed on the client's host servers. Whilst the most common software (programming language) deployed in a number of CMS is PHP, there are CMS' based on other programming languages such as Asp.net, Perl, Python, Javascript or Ruby. Factoring server/hosting issues, expected traffic and use case would be critical to the choice of CMS and webhost provider.

Other considerations would include basic understanding of relevant applications such as html5, Css, Php and some Jquery or javascript. Most CMS platforms do not necessarily require high end programming skills. Critical to deploying most server based CMS' is understanding of databases such as MySQL, MariaDB, SQlite etc. The newcomer however would be better placed where there is some understanding of computer systems and are resourceful in 'finding out' where there are gaps in knowledge.

My earlier article ['Getting Drupalysed' - Toolkit for the newbie](#) explored the essentials to deploying Drupal on a remotely hosted website.

Aside

My experience of Drupal CMS began over a decade ago when a colleague wanted a Website for their clinic which has since grown to become a multicentre Hospital

Group. I initially deployed a Wordpress based website, but soon realised the contrains could be better managed with a CMS with more scope to accommodate the demands of a busy setup as that use case. When the opportunity arose, and after learning the ropes with Drupal 7, I encouraged the client to consider a move to Drupal CMS.

Why Windows IIS

Working in a development environment is often necessary especially in the process of learning a new CMS. It saves the cost that may otherwise be incurred in securing a domain name and also hosting the site on a remote server before one is actually ready.

There are other advantages in addition to test-running the would-be website.

The development server allows new modules to be tested on the current Drupal version before it is deployed on the production server.

In my early days of web development I did experiment with XAMP and briefly MAMP on Windows. This required minimal adjustment as these applications are designed to embed APACHE in Window based computers. However I often felt my learning was incomplete as I was yet to effectively unleash the power of [Windows Internet Information Service (IIS)](#) which I remained curious about.

Setting up drupal on local IIS server environment.

Need:

- Latest version of Drupal 8 => 9
- PHP, MySQL and IIS
- Microsoft Web platform installer -
- Other useful programmes - Php manager, composer
- One good IDE - atom or vscode etc
- Or just a texteditor like sublime, notepad plus
- Some familiarity with CLI such as Git Bash or other terminals

How?

1. Configure windows IIS by switching on network parameters in programme features
2. Install latest PHP for windows NTS x64 vs x32 version depends on your windows system. If you are a developer you should be using a x64 computer in most cases. Make sure you note where PHP is installed in C: drive C:\Program Files(86)\PHP7 .... or C:\Program Files\PHP7 .... . Go to windows environment variables and add the php drive info
3. Download and install MySQL - get the free community version from oracle website 'https://dev.mysql.com/downloads/file/?id=499256'. Then also get the installer which makes deploying the mySql server on your PC a breeze. Download the installer via this link ' https://dev.mysql.com/downloads/windows/installer/8.0.html'
4. Setting it up the first time can be tricky but follow the instructions. The installer would prompt you to select what version of MySQL you like to down load including the latest versions. Choose one, preferrably the latest unless of course you are reinstalling a previous version suitable for your needs.
5. Set up the login parameters as per prompts.
6. Download Php manager for IIS and set up PHP to point to above drive. Activate the PHP extensions (about 40).
7. Download and install Composer. Getting composer installed correctly is critical. Composer makes setting up drupal a breeze as it installs, updates and configures the server environment to work with php automatically.

Extensions

Because composer takes over php settings, it will choose its required php extensions and set it up in php.ini file. These extensions will be in keeping with 'apache' and will not have the 'php_[file_name].dll prefix. This however opens you to errors as windows also needs some of its on php_extensions with the prefix on IIS for drupal. php_openssl.dll and php_pdo_mysql.dll therefore needs to be enabled.

For this reason it is good practice to dedicate a separate PHP folder e.g in C:\PHP separate from PHP in C:\Program Files\PHP, used for CMS such as drupal whose ext requirements are different from composer.

It is also possible to enable the 'composer required PHP' extensions in the same PHP folder used by drupal in anycase as long as one is aware of this possible pitfalls.

Initial Checks

Might help prevent deployment errors

- which php version
- which composer version and location of its php (compatible with IIS installation location)
- php -v
- composer -v
- Extensions!

Composer required php extensions

If running composer for the first time

curl

mbstring

openssl

pdo_mysql

IIS (windows) required php extensions

May be enabled using IIS Manager or in PHP.ini

php_mbstring.dll

php_openssl.dll

php_pdo_mysql.dll

php_mysqli.dll

php_opcache.dll

php_openssl.dll

php_gd2.dll

php_gmp.dll

php_wincache.dll

zend_extension=php_opcache.dll; Determines if Zend OPCache in enabled

opcache.enable=1

opcache.memory_consumption=128

opcache.interned_strings_buffer=8

opcache.max_accelerated_files=4000

opcache.revalidate_freq=60

opcache.fast_shutdown=1
Other PHP.ini Settings
Best check and update php.ini file
default_socket_timeout = 60
error_log = "C:\WINDOWS\Temp\php_errors.log"
upload_tmp_dir = "C:\WINDOWS\Temp\"
cgi.force_redirect = 0
cgi.fix_pathinfo = 1
fastcgi.impersonate = 1
fastcgi.logging = 0

Site path or 'docRoot' and Web.config

Another challenge with setting Windows IIS server to run effectively with Drupal 8 and later versions, is linked mainly to configuring the physical path and the localhost site address correctly. This requires using the application host facility in IIS.
The default application host settings makes it easy to identify the document root, virtual path and the site link. Geting this right can save one a tonne of hardship.

Windows server demands the web.config file is set correctly to identify the defaultDocument 'webroot' file typically index.php in drupal sites. It also requires the correct server 'handlers' associated with the server are registered. This was a challenge intially. I previously would deactivate the web.config file and the Drupal site still ran - installed and function okay on IIS. The draw back with this is that one may be unable to deploy the wealth of server resources that allow functional website - emails via IIS etc. I did resolve this config issue after a few research on its setup. The main idea is to get the default Document settings right and it should work.
(Link to image file on my set up).

Permissions

Granting write access to drupal during installation enables creating or accessing the files folder and the settings.php file.
Setting permissions in linux servers is usually straight via cml. In windows IIS server this is different and made user-friendly via the GUI. You typically would right-click on the file or folder, go to properties and select the security tab to access the permissions control interface. I have previously not had difficulties with this step as I

understood this process. Recently however after setting up IIS on a re-installed windows laptop I got stymied as I ran instal. Later did I realise the required permissions was for the IURS user and not the IIS_IURS user which I had been allowing.

After resolving this the installation continued smoothly.

Clean URLs with IIS

One final challenge is to resolve the 'Clean URL' settings to ensure all pages are accessed without '404' errors.

http://www.drupal.org/node/3854

To do this, the URL rewrite module needs to be installed on ISS by launching the Web Platform installer(WPI).

(If not already downloaded)
Open IIS manager and whilst on the main User account top folder (above Application Pools), have a look at the right far column in IIS Manager and click on "Get New Web Platform Components" which opens up a new browser tab to ' https://www.microsoft.com/web/downloads/platform.aspx'. Download the Web Platform Installer which would be added on to the last section "Management" in the Features view of IIS manager.

Now Click on the WPI nd searching for the URL Rewrite module.

Once installed, click and edit the icon to add the default domain name which will ensure the links all deploy with clean URLs.

Setup and Enabling write on files and /tmp folders

Finally enable write on files folder and set up the /tmp folder for drupal to access via IIS. This is because Drupal is unable to deploy this automatically in an IIS environment.

Set up /tmp folder outside drupal root.
I achieved this simply by creating a 'tmp' folder in the project root, that is within 'site_dir'.

Post installation tasks

Install Drush via Composer - composer require drush/drush

Consider installing [drush launcher](#) as this does make the deployment of drush much easier - updating, enabling, clearing cache or installing user accounts.

Extend drupal functionalities by [installing required modules and dependencies via Composer](#)

*For more information on [installing Drupal 8 or 9 on Windows](#)

Conclusion

Over the past few weeks, I had challenged a young enthusiast to deploy Drupal 8 on his laptop for the first time using IIS server. As this was done remotely by combination of phone or video-link, it did require motivation on his part. The reward was the successful installation and launching of his first ever website in far-away Nigeria!

It is probable many would be Drupalists may have been discouraged and turned away from this incredible CMS. Many seasoned developers indicated their frustration especially following the launch of Drupal 8, signaling the end of Drupal 7. With release of Drupal 9, which offers a seamless migration from Drupal 8, it is probable that advances in practice would position Drupal CMS in a better stead for the future

For new takers, the ability to experience Drupal CMS in a safe and unfettered development environment is perhaps essential to improve confidence and utility. Windows IIS remains the most ubiquitous resource for many who may never venture into the world of linux based computers or might find XAMP/WAMP/MAMP too much of detractor from the learning experience

-

Tags
[Web CMS](#)
[Drupal](#)
[IT](#)
Section
[Web developer](#)
➕

[View PDF](#)