

Installing Pydio on Windows Server 2012 R2 / IIS 8.5 – Update May 2017 (Pydio 7/PHP 7/x64)

<http://allandynes.com/2017/05/installing-pydio-7-on-iis-8-5-pure-64-bit>

This guide is assuming a vanilla install of Server 2012 R2 and lists all prereqs to get Pydio community edition up and running with IIS 8.5 / Server 2012 R2. I have found that none of the current guides cover everything 100% so through lots of testing and looking at all the other guides I wrote this one. It's making a couple assumptions which I think make sense for the average Windows based server:

- You are starting with a fresh Server 2012 R2 install. It is setup how you want it setup (named, domain joined, etc) and all windows updates are done.
- MySQL will be used for the database locally.
- LDAP/AD authorization will be used for all users.

Adding the server roles

We will need to install some roles and features for Pydio and PHP to work properly and also some pre-reqs for additional software we will be installing later. Start up the Server Manager if it didn't open on its own at login and click **Manage** → **Add Roles and Features**.

- At the Before You Begin screen click **Next**
- Leave the default of Role-based or feature-based installation and click **Next**
- The local server should be selected by default. Click **Next**
- In the list select Web Server (IIS)
 - o For the Add features required box that pops up click **Add Features** then **Next**
- At the Select features screen check off .Net Framework 3.5 Features (required by the PHP manager later) and **Next**
- At the Web Server Role (IIS) screen click **Next**
- At the Select role services screen leave all the defaults checked and then expand Application Development and check "CGI" then **Next**
- At the Confirm installation selections you will most likely get a yellow warning at the top about specifying an alternative source to download the .Net 3.5 files from. Click **Specify an alternative source path** at the bottom and enter in the path to these files. If you are using a Server 2012 DVD it would be D:\Sources\SxS\ where D: is your DVD drive. Once entered click **OK** then **Install**.
- Once installed click **Close**.

Next check Windows Updates as there will be updates to the .Net 3.5 framework that was just installed. Install all updates then reboot before the next step.

Installing required software

You can install some of these through the built in Microsoft Web Platform Installer 5.0 (Web PI) but after doing so many times I have found it easier to download and manually install the files I need. The Web PI also seems to have some outdated files and I rather have the most up to date ones as possible.

First download PHP 7.0.*: <http://windows.php.net/download/> . You want the latest one labeled “VC14 x64 Non Thread Safe”. Extract the zip file to C:\Program Files\PHP. PHP 7.1 does not currently work with Pydio....don’t use it.

PHP Manager is next so download the x64 version: <https://phpmanager.codeplex.com/releases/view/69115>. It still lists as for IIS 7 but there have been no changes and works fine on IIS8/8.5.

PHP requires the Microsoft Visual C++ 2015 Redistributable and MySQL requires 2013. Download the 64-bit version of 2015 here: <https://www.microsoft.com/en-us/download/details.aspx?id=48145> and the 64-bit version of 2013 here: <https://www.microsoft.com/en-us/download/details.aspx?id=40784> then install both.

Download MySQL here: <https://dev.mysql.com/downloads/windows/installer/> Grab the offline msi as it contains both the 32-bit and 64-bit install. Do a Custom install and select MySQL Server x64 under the MySQLServers group and then MySQL Workbench x64 under the Applications group. Click Next and Execute then Next a couple more times. Once it’s done select the “Server Machine” config type then Next. Enter in a root password and write it down as you’ll need it later to create your database. Click next through the remaining screens then Execute, Finish, Next and uncheck Start Workbench then Finish again to finish the install.

You will need the Microsoft URL Rewrite 2.0 module for public links to redirect correctly. Get that here: <http://go.microsoft.com/fwlink/?LinkID=615137> and install it

Lastly download and install the WinCache extension:

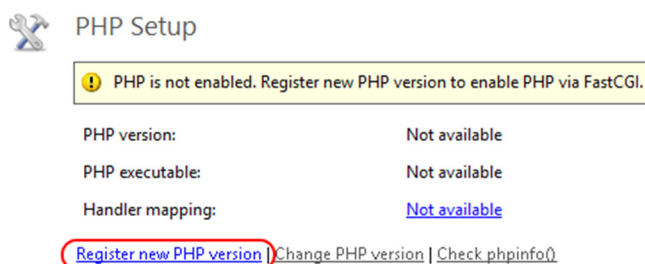
<https://sourceforge.net/projects/wincache/files/development/wincache-2.0.0.8-dev-7.0-nts-vc14-x64.exe/download> and run that. It will extract some files to the directory of your choosing. Copy the file php_wincache.dll to the C:\Program Files\PHP\ext directory. This will help speed up PHP on your Windows server dramatically.

Download Pydio

Download the latest community version of Pydio from <https://pydio.com/en/get-pydio> and get the “Pydio Community (Zip)” (or try <https://pydio.com/en/download-community-zip-archive> directly). Unzip the zip file to C:\inetpub\wwwroot\pydio or whatever name you choose. For my server I called the folder “cloud.mydomain.com” which I will be using throughout these instructions.

Configuring PHP

There are some settings to change in PHP to make Pydio happy. Startup IIS Manager then click on your server in the tree on the left. You will probably get a prompt about the Microsoft Web Platform. Click the “Do not show this message” box and then **No**. Double click the PHP Manager and you should have a yellow warning that PHP is not enabled. Click “Register new PHP version”:



Click the browse button (three dots) and select C:\Program Files\PHP\php-cgi.exe then **OK**. Now click the Set Runtime Limits link. Here is one thing that will limit your ability to upload large files. Personally I have my “Maximum POST size” and “Upload Maximum File Size” both set to 512M, my “Maximum Input Time” set to 300, and my “Memory Limit” set to 512M. Set as appropriate for your environment then click the **Apply** button in the top right then **Back to main page**.

Let’s turn on a bunch of extensions and turn one off if it’s enabled. Click “Enable or disable an extension” then in the Disabled list find php_ldap.dll which we will need for LDAP/AD authentication and also php_com_dotnet.dll for PHP command line to work and enable both of those. If you will be using the email function find php_snmp.dll and enable that and php_exif.dll if you plan on handling images (probably). For non-english support enable the php_intl.dll extension. And lastly enable php_opcache.dll which is supposed to speed up PHP processing. Now find php_mysql.dll and disable it (we are using php_mysqli.dll). Click **Back to main page**.

While still in the PHP Manager click “Manage all settings” then edit the following settings:

- upload_tmp_dir: Change this value to “C:\inetpub\temp”.
- session.save_path: Change that to "C:\inetpub\temp" also.
- output_buffering: Change it from the default of 4096 to “Off”.
- SMTP: Change to your email server if you are using this.
- smtp_port: Change if you use something other than the default of port 25

Click **Back to main page**.

Last thing to do is change the opcache extension and configure it. Click the link for your PHP configuration file, should be php.ini. Scroll to the very bottom and find the line that says “extension=php_opcache.dll”. Change this to “zend_extension=php_opcache.dll”. Then add the following settings after this line:

```
opcache.enable=1
opcache.enable_cli=1
opcache.memory_consumption=128
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=4000
opcache.revalidate_freq=60
opcache.fast_shutdown=1
opcache.save_comments=1
```

Save and close the configuration. Once all these changes are done PHP should be configured correctly for Pydio

Creating the Application Pool

You should still be in the IIS Manager. On top left menu select “Application Pools” then right click it and select “Add Application Pool...”. For the name enter pydio and change the .Net CLR version to “No Managed Code” then click **OK**.

Creating the Site

On the left side highlight “Sites” then right click it and select “Add Website...”. Enter a name for your site, again I’ll be using cloud.mydomain.com to match my directory name. Click the “Select...” button on the right and select the “pydio” application pool you created in the last step then **OK**. Click the browse button for the physical path (three dots) and select your directory under C:\inetpub\wwwroot\ (pydio directory) and then **OK**. Under the binding enter the host name that will be used. Again for mine I am entering cloud.mydomain.com. Click **OK**.

Once created expand out “Sites” and select your pydio site. Double click the “Authentication” button then right click “Anonymous Authentication” and select “Edit...”. Check off “Application pool identity” and click **OK**.

Setup Folder Permissions

Time to set some folder permissions for Pydio. Navigate to your C:\inetpub folder. First right click the temp directory and go to **Properties**. Click the **Security** tab then click **Edit**. Click **Add** then under **Locations...** make sure the server is selected at the top. In the object names field enter “iis apppool\pydio” then **Check Names**. It should resolve to just “pydio”. Click **OK** then give this user Modify writes to the temp folder and click **OK** twice.

Go up a directory and then navigate to your wwwroot directory which should be c:\inetpub\wwwroot. From here right click your Pydio folder and select **Properties** then click the **Security** tab. Click the **Advanced** button at the bottom then **Disable Inheritance**. Select to **Convert inherited permissions into explicit permissions on this object** then check the box marked “Replace all child permission...” then **OK**. You will get a warning. Click **Yes**. Click the **Edit** button back on the Security tab, find the “IIS_IUSRS” group, and **Remove**. Now click **Add** and add the pydio application identity (iis apppool\pydio) as you did above for the temp directory. Leave the default rights of read and no write access then click **OK** twice.

Now we have to give the pydio application pool identity write access to the data directory. Go into pydio folder, right click the data directory, and edit your pydio identity to have Modify rights by clicking **Edit** then adding the modify right for the pydio user. Do the same for the .htaccess file also.

Last thing to do is go into the root of your Pydio folder and rename the “web.config.sample” to just “web.config”. The Pydio directory should be ready.

Create the pydio database in MySQL

Start up the MySQL command line. Enter in the root password you wrote down earlier and then at the command prompt enter the following to create a database and a user for the database:

```
CREATE DATABASE pydio;  
CREATE USER "pydiouser"@"localhost" IDENTIFIED BY "mypydiopw";  
GRANT ALL PRIVILEGES ON pydio.* TO "pydiouser"@"localhost";
```

Type exit to close the command prompt.

Enable PHP to run from the command line

Pydio can use the PHP command line to offload some tasks. We have already enabled the PHP extension needed for this and now need to add PHP to the systems path variable and registry. Right click the start button and select System then on the left click "Advanced system settings". At the bottom click "Environment Variables...". At the bottom under "System Variables" find the "Path", highlight it, and click "Edit". Scroll to the end and add a semicolon then the path to your PHP install which if you have been following along is "C:\Program Files\PHP". Click **OK** and then find "PATHEXT", highlight it, and click "Edit". Scroll to the end and add a semicolon then ".PHP" and click **OK**. Click **OK** two more times to exit the System Properties page then close the System screen.

One last thing to do is add a PHP association to windows. Save the following contents to a file called "PHPCLI.reg" to your servers desktop:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\.php]
@="phpfile"
"Content Type"="application/php"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\phpfile]
@="PHP Script"
"EditFlags"=dword:00000000
"BrowserFlags"=dword:00000008
"AlwaysShowExt"=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\phpfile\DefaultIcon]
@="C:\\Program Files\\PHP\\php-win.exe,0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\phpfile\shell]
@="Open"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\phpfile\shell\Open]
@="&Open"

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\phpfile\shell\Open\command]
@="\"C:\\Program Files\\PHP\\php.exe\" -f \"%1\" -- %~2"
```

Once saved double click the file to import the information into your registry and you should now be able to execute any PHP file from any path without having to call php.exe directly. Reboot to let the new settings take effect.

Start the Pydio Install Wizard and finish the install

All our setup should be done. Browse to your pydio installation using the full URL, in my example its <http://cloud.mydomain.com/>. You will be presented with a check of the requirements for installation of pydio which should have no errors after following the steps above although you might have a warning for SSL and possibly the PHP Opcode extension which can both be ignored. If you have any red errors then something is wrong and they need to be fixed before continuing, make sure you followed all the steps above correctly. If you get an Internal Server Error 500 make sure your PHP version and VC redistributables are all correct.

Click Continue to Pydio Installation. Choose your language and click Start Wizard. You may want to change the application title, I changed mine to "MyDomain Cloud Storage" and same with the welcome message. For the

admin login and password pick whatever you want, this will only be used for the initial setup as we will be switching to LDAP/AD for future access. Click the next arrow.

For configurations storage we will be using MySQL. For the host change it from localhost to 127.0.0.1. For the database name, user, and password use the information in the step above (pydio, pydiouser, and mypydiopw). For Use MySQLi check Yes. Click “Test Connection” and verify it works. If not make sure your MySQL database setup was correct. Once it works click the next arrow.

For the detected encoding you shouldn’t have to change anything. However for the detected URL you may have to switch the last slash to a forward slash instead of the backslash. Leave “Enable cache (recommended)” as **No** (It appears Pydio 7 uses the PHP_APC extension for this type of caching which doesn’t exist in a 64 bit version. Also the wincache extension should take care of this for the most part). Enable emailing if you plan on having that functionality and select “Mail” for the PHP Mailer. Put in an email account you want to send general notifications from. I use DoNotReply@Cloud.MyDomain.com. You can test this by clicking the “Try sending an email with the configured data” button but make sure you use your own email address when you test this. You can also select a default language if it’s other than English.

Click the **Install Pydio** button to apply all these settings. It should be able to write to the .htaccess file since you changed the security earlier. You should now lock that file back down. Go to C:\inetpub\wwwroot\pydio\ and edit the security settings on the .htaccess file and remove write access from the pydio application pool.

You should get a login page. Login as the administrator that you just defined.

Setting up LDAP access

In my scenario I will be setting up LDAP access for my users. I don’t want **ANY** local users. In fact after this step and a reboot of the server the default admin login will no longer work. After this we will lock down some of the interface so users can’t change their account (the info will come from Active Directory...we don’t want them to change anything).

You should be logged in as the administrator. Click the admin drop down in the top right then **Settings**. Click **Application Parameters** then **Application Core**. Double click **Authentication** and change as follows:

Generic Features – Store Credentials in Session: If you are going to use SMB to access another Windows server file shares then switch this to Yes.

Login Form

- Secure Login Form: I recommend switching this to Yes
- Enable Forgot Password: Leave on No since we are using LDAP/AD

Main Instance (I’m only listing one that I am changing. Use defaults for the rest)

- Instance Type: LDAP/AD Directory
- LDAP URL: enter your primary domain controllers FQDN such as DC.mydomain.com
- Protocol: leave on standard unless you are using LDAPS
- LDAP Port: 389 for standard. 636 for LDAPS

- LDAP Bind Username: This is the full path to your user. If you create a service user called “LDAP Lookup” in a OU called “Service Accounts” for your domain “mycompany.com” it will look something like “CN=LDAP Lookup,OU=Service Accounts,DC=mycompany,DC=com”
- LDAP Bind Password: The login password for the above user
- People DN: The OU to search for users such as “OU=Users,DC=mycompany,DC=com”
- LDAP Filter: Filter to use against your users. You can use the default or use mine, it works well to pull all users that have an email address assigned so it filters out most of my service accounts: “(&|(objectClass=person))(mail=*)”
- User Attribute: The attribute that the users will use to login. I default to sAMAccountname, you might use mail.
- Groups DN: The OU to search for groups you want available for assigning to workspaces.
- Group Attribute: The attribute for listing the group names. I use sAMAccountName.
- Role Prefix (for memberof): The prefix that will be used on the roles that are pulled from AD. I recommend using “ldap_” so you can easily tell where they came from.
- LDAP Attribute: I recommend adding three. The first will map your users real name to the user display name, the second will map their email address, and the third will be used for the AD group membership
 - o LDAP Attribute: displayName / Mapping Type: Plugin Parameter / Plugin parameter: core.conf/USER_DISPLAY_NAME
 - o LDAP Attribute: mail / Mapping Type: Plugin Parameter / Plugin parameter: core.conf/email
 - o LDAP Attribute: MemberOf / Mapping Type: Role Id / Plugin parameter: (Blank)
- Fake MemberOf: Switch to No
- Search Users by Attribute: I usually use “displayName,givenName,cn”

Hit Save then scroll back down to “Test User”, enter a known user ID or email if that’s what you chose for user attribute, and click “Try to connect to LDAP”. It should be successful. If not figure out what is wrong in your LDAP/AD config before proceeding.

We cannot publicly share files until a secondary form of authentication is setup. Scroll to the bottom of the Authentication screen and under “Multiple Instances Mode” pick **Master/Slave**. For Cache master users select No. Under “User Listing” switch to **Master Only** then for “Secondary Instance Type” select **DB Auth Storage**. Click **Install SQL Tables** then **Save**.

VERY IMPORTANT: Since I am only using LDAP/AD authentication at this point, and the secondary is only for the public links to work, the default admin login will not work once you log out and your session is destroyed. This is how I want my setup as I rather control all users through AD and not have a second set of users in Pydio. So the next thing to do is assign one of your AD users an administrator so you can get back in after you log out of the default admin account.

Under “Workspaces & Users” click “People” on the left and you should have your user list from AD. Give it a minute to pull users and if it seems hung up click the refresh button. Select one of your users in the list, probably yourself, and then click Edit. Under “Special Profile” select “Administrator” then **Save**. Open a **different** browser on your system or connect from another system, navigate to the site, and verify your AD user can login and has administrative rights with a “My Settings” option.

Setup our default Role to lock down the user

Now to setup the default role and lock out some stuff I don't want my users getting into. On the left under "Workspaces & Users" click "Roles". Double click the "Root Group". Enter in a country and language. Click on the ACL and for my users I deny the default Common Files (I'll create my own later), My Account, and Settings. I set My Files, Home, and Shared Files as read write. Click the **Actions** tab. Now to lock the user out of the My Account screen since all our info is coming from AD. Select Conf.sql - switch_to_user_dashboard (My Account) – All Workspaces and **Add Action**. Click **Save** and the "My Account" option should not be there when a user logs in.

Enable the Command Line PHP

IIS doesn't seem to know how to use CLI PHP without its extension so we need to change that. While in Settings navigate to Application Parameters → Application Core → Pydio Main Options. Under Command Line change "Command-line PHP" to the full executable name of "PHP.exe" and then switch "Use COM class" to Yes. Personally while I'm in here I also disable Zip Creation by switching it to No because I don't want people downloading massive amounts of files as a zip. Click **Save** when done.

Scheduling Tasks

I use a workspace in Pydio that is actually a DFS replicated folder. The problem I have found is because files are being added to this workspace outside of Pydio that it isn't very keen on indexing it automatically. Now there is a scheduler built into Pydio but in my tests I simply couldn't get it to work reliably on IIS. You can however launch tasks directly from the command line for a host of things including indexing using a guide on the Pydio website (<https://pydio.com/en/docs/v5/command-line-version-server>). Here is how to do it

- Create a domain user that will be set as an administrator within Pydio. I called mine PydioServiceUser.
- Create a bat file on your web server to run a command. I call mine "PydioScheduledTasks.bat"
- In this bat file call PHP and the Pydio. For example this command will run an index on the repository specified:

```
"C:\Program Files\PHP\php.exe" "C:\inetpub\wwwroot\cloud.mydomain.com\cmd.php" -a=index -r=5b7536hb64461530c9be235 -u=PydioServiceUser -p=MyPassword
```

- To get the repository ID you can either search through the database or go into the actual Pydio scheduler, setup some task, and save it. It will list your repository ID
- Now go into the standard Windows Task Scheduler and schedule the bat file. I schedule mine to run Daily at 12:05am, recur every 1 day, repeat task every 30 minutes for a duration of 1 day.

There are a lot of things you can schedule but this is the only one I use.

Testing

Login as a non-administrator AD account and you should find your correct AD name pops up in the top right and there is no My Account option. Upload a file to your My Files workspace and then share it out. It should generate an "<http://cloud.mydomain.com/public/7adbe5>" link which if you paste into a different browser or another system should redirect to the file. Click the **Invite** button and type in your email address and send. Test the link in your email also.

Some extra customizations

I have some customizations that work for me but might not for you. If you are happy with your Pydio install and want to play then you should stop here. However I have found for a business certain things don't make sense to allow and have disabled them. Here are some of them and how to disable.

Under "Workspaces & Users" → "Roles" → "Root Group" → Actions:

- No downloading files chunked: access.fs – download_chunk – All Workspaces
- No copying files: access.fs – copy – All Workspaces
- No moving files: access.fs – move – All Workspaces
- No CHMOD: access.fs – chmod (File Permissions) – All Workspaces

I don't want users sharing files internally. I change that under "Application Parameters" → "Feature plugins" → "Action plugins" → "Sharing Features"

Authorizations → Files: enable internal sharing > No

Authorizations → Folders: enable internal sharing > No

Under the same plugin I want to display the direct download link:

Link Generation → Display Direct Download Link → Yes

Lastly I do not want my users to have a recycle bin in their "My Files". That's removed by editing the "bootstrap_repositories.php" file under the conf subdirectory and changing the "RECYCLE_BIN" setting from 'recycle_bin' to an empty string (").

After all these customizations the site works how I need it to. Modify for your own use and play with it.

Remember that some of these changes need the plugin cache cleared before they take effect or reboot the server which also seems to do the trick.