

GoDaddy guide to cPanel + WordPress

By Topher Derosia, Saurabh Shukla, Andy McIlwain



GoDaddy guide to cPanel + WordPress

By Topher Derosia, Saurabh Shukla, Andy McIlwain

Copyright 2017 GoDaddy Operating Company, LLC. All rights reserved.

Table of Contents

Using cPanel:

Overview of the cPanel features

[Navigating cPanel](#)

[Section headers](#)

[Search filter](#)

[Managing user accounts](#)

[MySQL accounts](#)

[FTP Accounts](#)

[Email Accounts](#)

[Domain Names](#)

[Other domain options](#)

[SSL/TLS](#)

[Final thoughts...](#)

[Related reading](#)

Installing WordPress on cPanel

[Installing WordPress manually](#)

[Logging into your GoDaddy account](#)

[Opening cPanel](#)

[Create the database](#)

[Create the database user](#)

[Add the user to the database](#)

[Create FTP user](#)

[Download WordPress](#)

[Unzip WordPress](#)

[Upload the WordPress files](#)

[Using the WordPress installer](#)

[What if the installer doesn't work?](#)

[Installing WordPress via cPanel](#)

[Final thoughts on installing WordPress with cPanel](#)

[Related reading](#)

Managing WordPress with cPanel

[Manually upload plugins and themes](#)

[Running backups](#)

[Restoring backups](#)

[Accessing error logs](#)

[Related reading:](#)

[Troubleshooting with WP_DEBUG](#)

[Using phpMyAdmin with your database](#)

[Exporting your database with phpMyAdmin](#)

[Importing a database SQL file with phpMyAdmin](#)

[Final thoughts on managing WordPress with cPanel](#)

[Related reading](#)

Advanced cPanel features for WordPress

[WordPress privacy and security with cPanel](#)

[Directory privacy & security](#)

[Preventing bandwidth abuse with hotlink protection](#)

[Blocking malicious connections with IP Blocker](#)

[Running commands remotely \(SSH access\)](#)

[Basic shell commands](#)

[cPanel Cron Jobs and WordPress](#)

[WP-Cron vs Linux Cron](#)

[Switching WP-Cron to cPanel cron jobs](#)

[Switching to cron jobs](#)

[Scheduling backups](#)

[Final thoughts on advanced cPanel features](#)

[What now?](#)

Using cPanel.

Using cPanel: Overview of the cPanel features

By Topher Derosia

There's a lot more to running a WordPress site than just WordPress itself. Websites need domain names, SSL certificates, databases, user accounts, and sometimes even email, if it's not hosted elsewhere

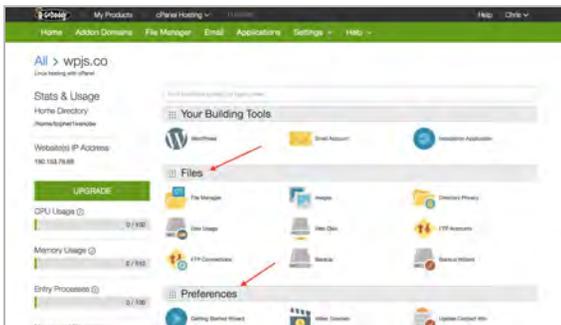
Thankfully cPanel can help with all of those things. So let's start with finding our way around. (Note: We're using GoDaddy's cPanel as the example for this guide. Screenshots were taken from both [Business Hosting](#) and [VPS](#).)

Navigating cPanel

Using cPanel is a bit like riding a bike. It's hard to get going, but once you get the hang of it, you never forget.

Section headers

cPanel is neatly organized. Each tool is grouped with other tools of the same type under section headers. Tools for working with files, for example, can be found in under the *Files* header.

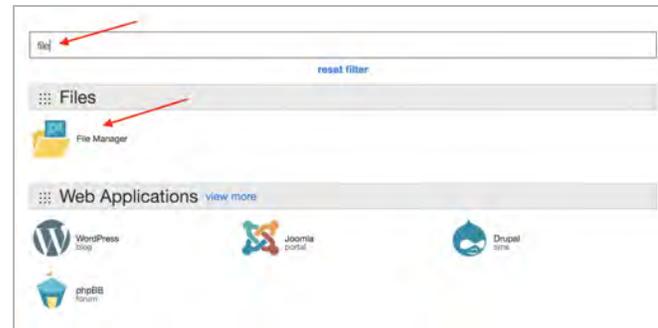


Search filter

Browsing for tools is great, but there's a faster way to find what you're looking for. By far my favorite way to find something in cPanel is the search filter.

At the top of the main cPanel screen is a search field. When you start typing, tools on the page start disappearing until all you're left with is the one that matches your search query.

Here's an example where I typed in "file":



The search results appear instantly. You're not waiting for the page to refresh. As your search changes, the visible results will change in real-time as well.

Managing user accounts

There are a number of different user accounts that need to be created for installing and running WordPress. They might all belong to the same person, but they're different accounts for different tools. (In some cases, WordPress itself is the user.)

Here's how you can manage them all in cPanel:

MySQL accounts

WordPress uses a MySQL database to store data like post content, WordPress author information, and plugin or theme settings.

To write to the database, WordPress needs credentials: a username and a password.

Tip: For added security, consider creating separate credentials for WordPress and each person that needs access to the database. That way you're not all sharing the same username & password.

First, we find the MySQL section in cPanel with the search filter:



Searching for "mysql" in cPanel

On the next page, you'll find a section for adding a new user:

A screenshot of the "MySQL Users" section in cPanel, specifically the "Add New User" form. The form has three input fields for "Username", "Password", and "Password (Again)". Below these is a "Strength" indicator showing "Very Weak (0/100)" and a "Password Generator" button. At the bottom left is a green "Create User" button.

Adding a new MySQL user in cPanel

And then below that, you'll find a section for editing existing users:

A screenshot of the "Current Users" section in cPanel. It shows a table with two columns: "Users" and "Actions".

Users	Actions
i3686432_wp1	Change Password Rename Delete
i3686432_wp2	Change Password Rename Delete
topher2	Change Password Rename Delete
wpjs	Change Password Rename Delete

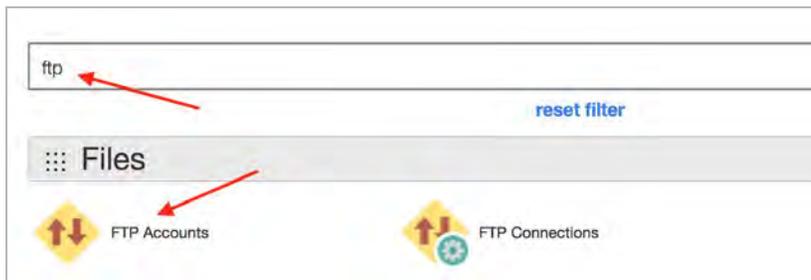
Viewing existing MySQL users in cPanel

Tip: If you change the password for the MySQL credentials used by WordPress, you'll need to edit the site's wp-config.php file and update the password. [See more details in the WordPress Codex.](#)

FTP Accounts

Use FTP accounts to connect to your server via FTP, SFTP, and SSH. For WordPress, you'll likely use an FTP or SFTP account when manually installing WordPress, or for performing administrative tasks that cannot be completed within the WordPress dashboard.

First, we find FTP Accounts with the search filter:



Searching for "ftp" in cPanel

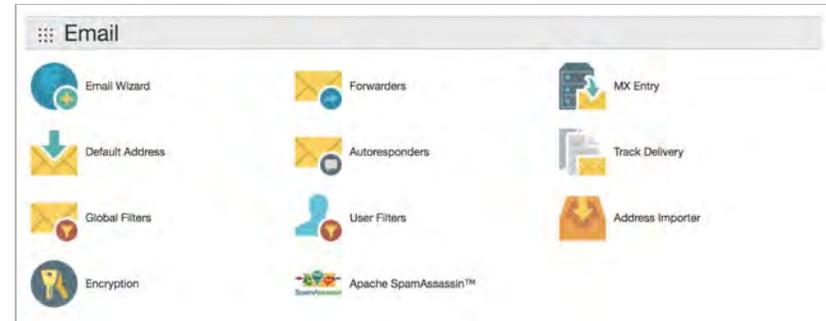
On the *FTP Accounts* screen you can create and manage all of your FTP and SFTP user accounts:



Viewing FTP Accounts in cPanel

Email Accounts

cPanel includes capabilities for managing email services like user email accounts, email forwarding, autoresponders, email tracking, and spam filtering. You'll find these tools under the *Email* section heading.



Email tools in cPanel

Tip: We generally recommend using a professional email service, e.g. [Office 365](https://www.office365.com/), for handling emails. A dedicated email service offers better performance, improved security, and additional features above and beyond what's available in a hosting account.

Domain Names

Every hosting account is associated with a domain name. (If you want to change the main domain name, cPanel makes it easy: Under *Preferences*, look for *Change Main Domain*.)



Note for GoDaddy customers: When you click *Change Main Domain*, you'll be taken out of cPanel and into GoDaddy's domain setup tool. Depending on the change you're making, it may take a day or so for the rest of the internet to catch up with your changes.

Other domain options

You can run multiple sites with different domains on the same cPanel account. These are called *addon domains*.

You can also create subdomains for your sites. Subdomains are like a prefix for your domain (in `http://cars.transportation.com`, 'cars' would be the subdomain). Subdomains are free to create. They're useful for sub-sites, e.g. membership portals, or for setting up test sites, e.g. `staging.yourdomain.com`.

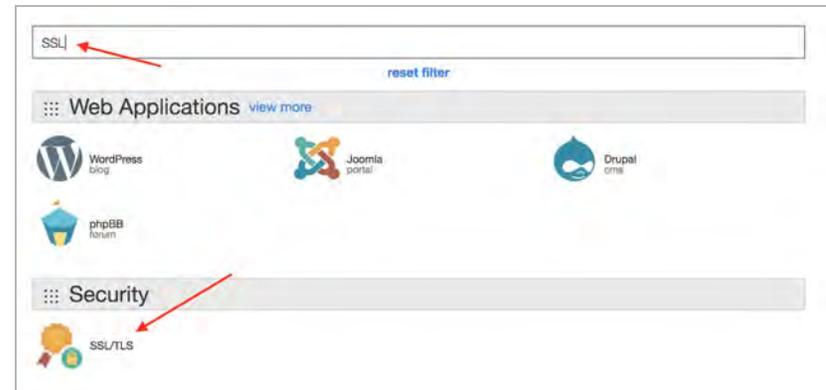
You can also create domain aliases so that two domain names go to the same website. This is especially useful if you register the same domain with multiple extensions like `.com`, `.net`, `.org`, `.biz`, and so on. You can make all of these domains point to your site.

SSL/TLS

SSL/TLS is a security protocol that encrypts web traffic between your server and the end user. When a site has SSL enabled, the URL will include "https://" and you may see a lock icon in the address bar.

HTTPS and SSL are a must-have for modern websites. [Google Chrome will even display a warning that a site is insecure if it's not using HTTPS.](#)

Using cPanel on GoDaddy, the SSL setup process is very straightforward. Under *Security*, look for *SSL/TLS*:

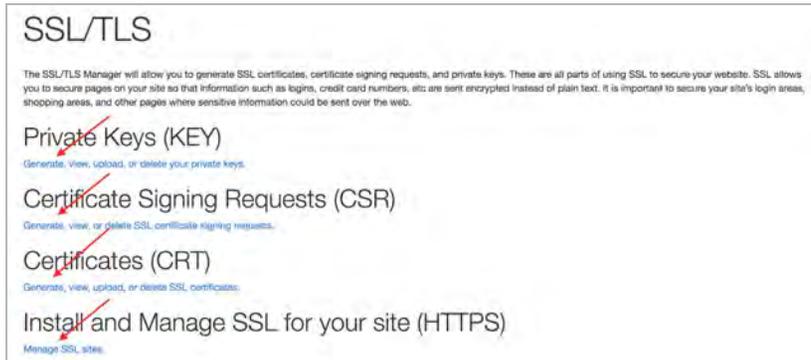


SSL/TLS in cPanel

Once you click the icon you'll see a page with 4 tasks on it. You need to accomplish each task:

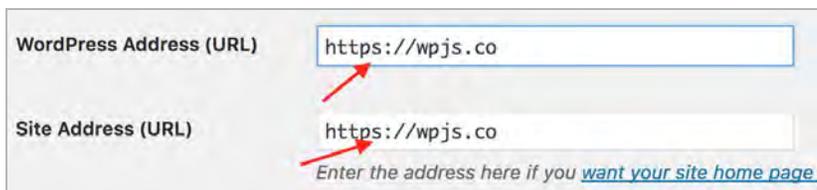
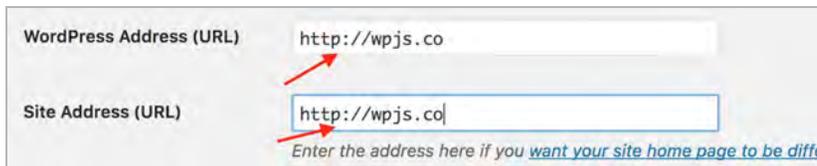
- Create Private Keys
- Create a Certificate Signing Request
- Get a Certificate
- Install the certificate

Click through each step and follow the prompts:



SSL/TLS configuration in cPanel

Once your SSL certificate is installed in cPanel you'll need to update your WordPress settings. Log in, and in the sidebar, click *Settings*. Update your *WordPress Address* and *Site Address* fields, replacing `http://` with `https://`, as shown below.



WordPress URL settings

Tip: The [Really Simple SSL plugin for WordPress](#) is incredibly useful for converting an existing website to HTTPS.

Final thoughts...

We covered quite a bit of territory in this post, and still only scratched the surface of what you can do with cPanel. Explore it all - there are some gems that can really enhance your WordPress site.

Related reading

- [What is SSL?](#)
- [Does your website need an SSL certificate?](#)
- [Paid and free SSL certificates - what you need to know](#)
- [Choosing the right security certificate for your business site](#)

Installing WordPress on cPanel.

Installing WordPress on cPanel

By Topher Derosia

cPanel is a powerful tool that allows people who are not system administrators to perform many of the tasks that would otherwise require an administrator. One of these tasks is installing WordPress.

WordPress is famous for the “[five-minute install](#)”, but that doesn’t take into account the steps of setting up a database and installing WordPress onto the server. cPanel can make these tasks easy and accessible for the average person. Let’s look at how to do it.

Installing WordPress manually

WordPress needs two things to work. It needs the **WordPress software** for loading and presenting the site, and it needs a **MySQL database** to store the site’s configuration settings and content.

In this section we’re going to walk through a manual WordPress installation. We’re using GoDaddy for this, but these steps apply to most cPanel web hosts.

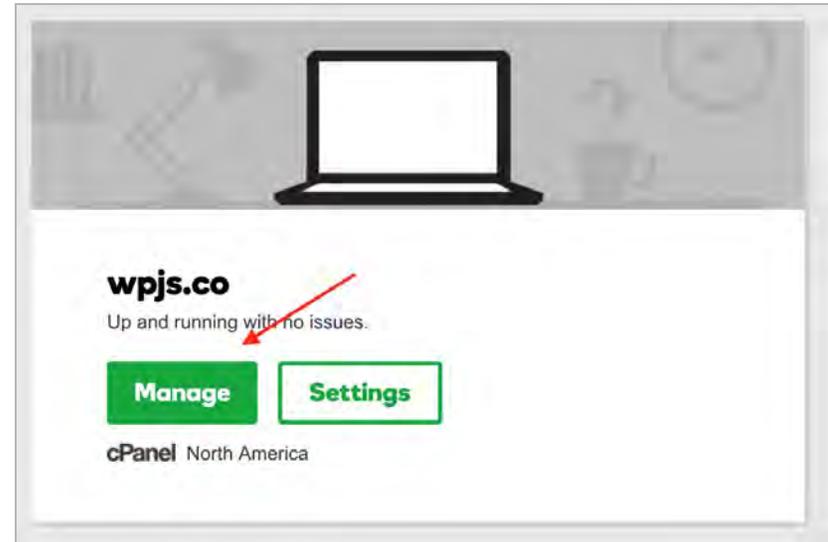
Logging into your GoDaddy account

The first thing we want to do is log into our GoDaddy account. Once you’re logged in you’ll see a list of your products. Next to Web Hosting, click *Manage*.

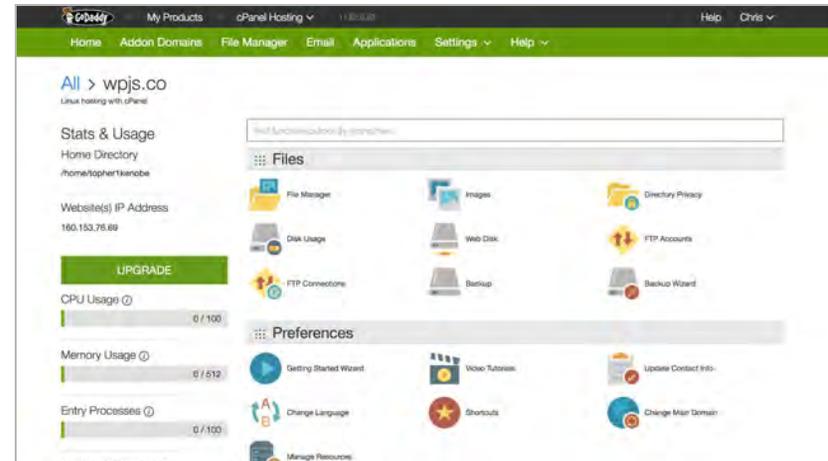


Opening cPanel

Each website hosted on GoDaddy will have its own *Settings* and *Manage* screen. To get to cPanel, click the *Manage* button.



This brings us into cPanel:

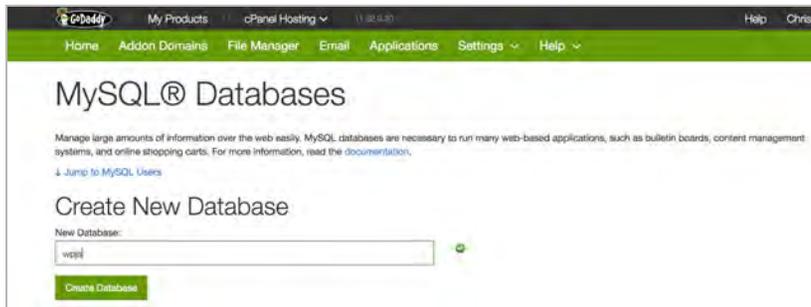


Create the database

Now that we're into cPanel, we'll need to set up the database that WordPress will use. Under *Databases*, click *MySQL Databases*.

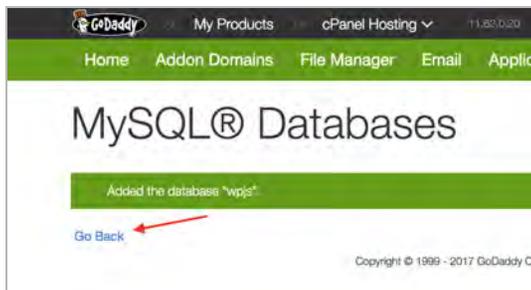


In the *New Database* field, enter a name. The name can be anything you want. My domain name is `wpjs.co`, so I'm calling my database `wpjs`. Enter a name then click *Create Database*.

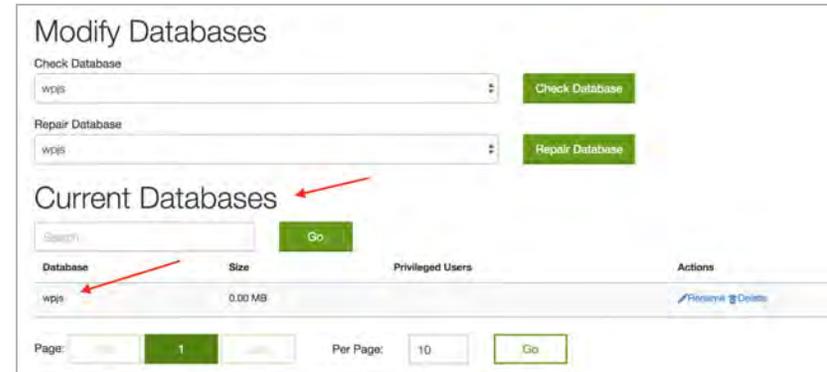


After you've clicked *Create Database*, you'll see a confirmation page indicating that the database has been created.

Return to the database management screen by clicking *Go Back*.



The database you just created will now be visible in the list of MySQL databases.



Create the database user

We've created our database. Now we need to create a user account. WordPress will use this account to access the database.

On the *MySQL Databases* screen, scroll down to *MySQL Users*.



The screenshot shows the 'MySQL Users' section with the 'Add New User' form. The form includes fields for 'Username' (wpjs), 'Password', and 'Password (Again)', each with a green checkmark indicating it's valid. A 'Strength' indicator shows 'Very Strong (100/100)'. There is a 'Password Generator' button and a 'Create User' button at the bottom.

The username is up to you. I recommend making it something similar to the database name so that you know that the two go together. (I'm using wpjs, the same name as my database.) Enter a strong password and click *Create User*.

Note: Make sure you remember your password - we'll need it when installing WordPress. You won't be able to view the password later; you will only be able to reset it.

Add the user to the database

Now we have a database and a user account. One problem: the user doesn't have access to the database yet.

On the *MySQL Databases* screen, scroll down to *Add User To Database*. Select the username you created from the list, and then select the database you created, and then click *Add*.



The screenshot shows the 'Add User To Database' form. It has two dropdown menus: 'User' (wpjs) and 'Database' (wpjs). There is a green 'Add' button at the bottom.

Next, you'll be asked to assign permissions to the new user account. Check the *Add All Privileges* checkbox, then click *Make Changes*.

A message appears confirming that the permissions were granted. Click *Go Back*.



The screenshot shows the 'MySQL® Databases' section with the 'Add User to MySQL® Database' form. A green message box states: 'User "wpjs" was added to the database "wpjs" with the requested privileges.' Below the message is a 'Go Back' link with a red arrow pointing to it. The footer shows 'Copyright © 1999 - 2017 GoDaddy Operating Co.'

That's it! Your database is ready to go.

Create FTP user

With the database set up, we need to get the actual WordPress software files onto your hosting account. For that, we'll use FTP.

From the main cPanel screen, under *Files*, click *FTP Accounts*.



On the *FTP Accounts* screen, go to *Add FTP Account*:

A screenshot of the 'Add FTP Account' form in cPanel. The form includes fields for 'Log In' (wpjs), 'Password', 'Password (Again)', 'Directory' (/home/topher1kenoba/public_html), and 'Quota' (2000 MB). A green progress bar indicates the password strength is 'Very Strong (100/100)'. A 'Create FTP Account!' button is at the bottom left.

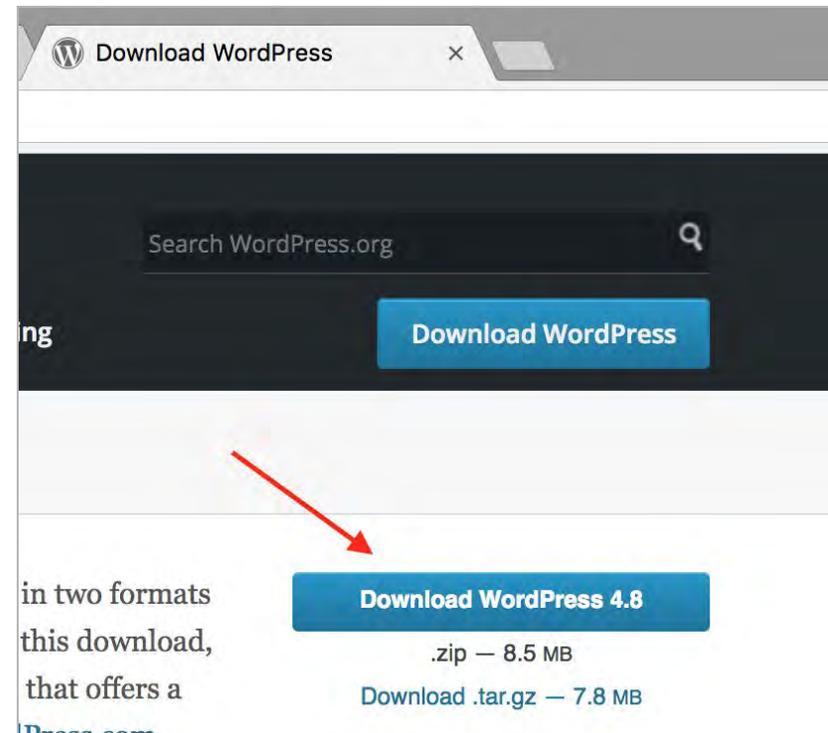
Under *Log In*, enter a username. Notice that your domain name is appended to it. In the adjacent screenshot you can see that I entered wpjs, but the *full* username is actually wpjs@wpjs.co.

Next, enter a strong password. Leave the path set to public_html. A file size quota is optional; I recommend leaving it set to Unlimited.

Click *Create FTP Account* to finish creating the account.

Download WordPress

Now we need to get the WordPress files. Go to <https://wordpress.org/download/> and click the *Download WordPress* button. This will download a copy of WordPress to your computer.



Unzip WordPress

Once you've download the WordPress .zip file, you'll need to extract (unzip) the files to a folder on your computer. You won't need to edit any of the files - they just need to be available for the next step.

Upload the WordPress files

You'll need an FTP client for this step. We recommend [FileZilla](#) - it's a free open-source app that works on Windows, Mac, and Linux.

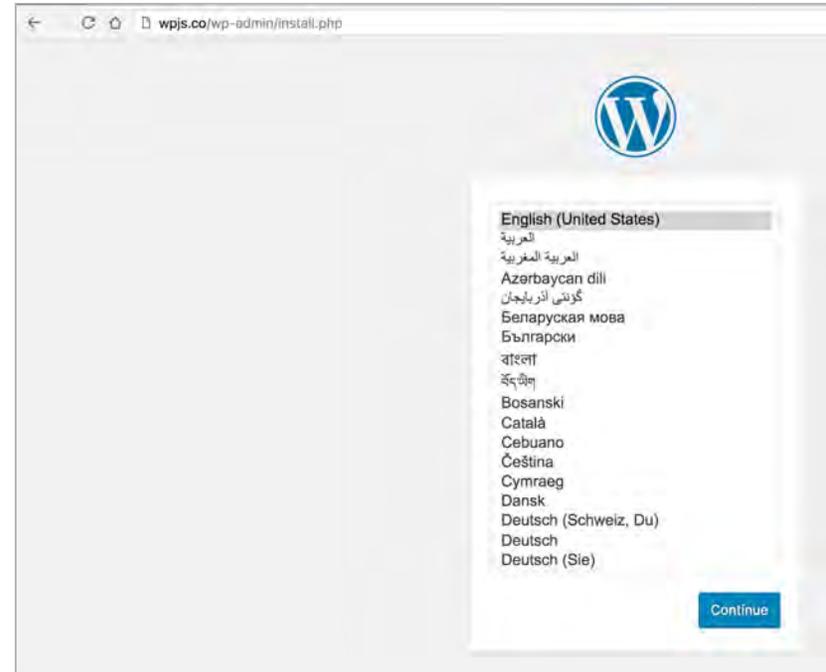
In cPanel, under FTP Accounts, look for the link to *Configure FTP Client*. This will bring up the credentials for your FTP account, plus the configuration files for recommended FTP clients (e.g. FileZilla).

Simply download one of those configuration files, double click, and your FTP client will connect to your server.

Once you're connected, upload the WordPress files that you downloaded earlier.

Assuming you're going to run WordPress on the domain, drag and drop the contents of the /wordpress/ folder from your computer to the /public_html/ directory on your server. (If you're installing WordPress in a different location, e.g. on a subdomain, upload it to the corresponding directory on your server.)

Once the upload is complete, open your browser and enter the URL of where you uploaded the files to. The WordPress installer should appear.

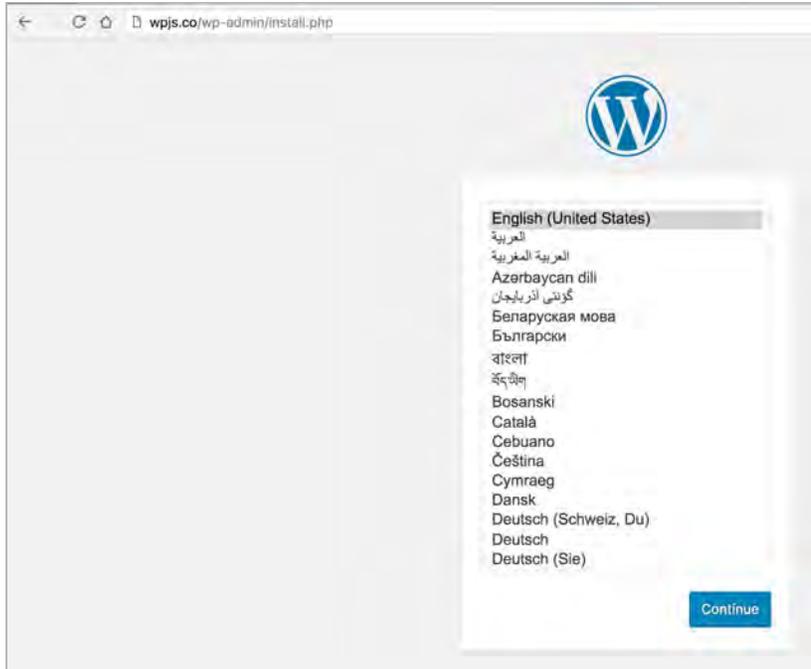


The WordPress installer prompting you to choose a language.

Using the WordPress installer

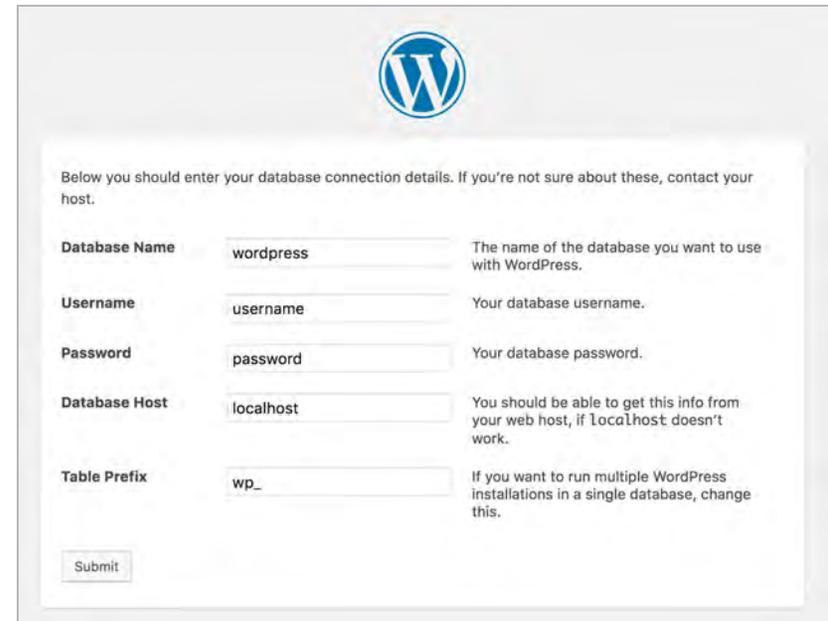
Welcome to the WordPress installer. The first step is to select which language you'll be using. (Note: This applies to the WordPress admin backend, *not* the content that appears on your website.)

Select your preferred language and click *Continue*.

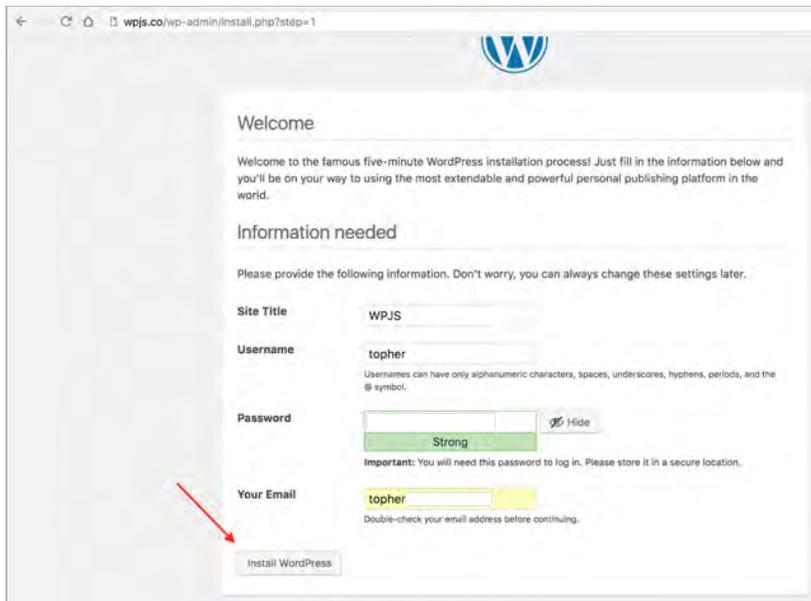


Now WordPress will ask you for your database connection information. These are the same MySQL user credentials that we set back when we created the database in cPanel.

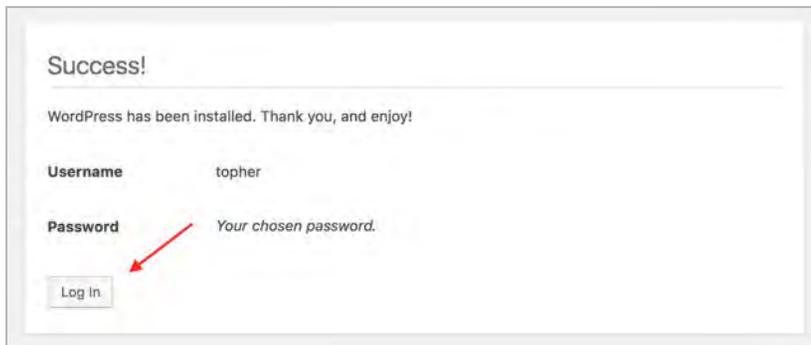
Note: If you want to use the same database for multiple WordPress sites, change the table prefix. Otherwise WordPress will get confused between the different installations all using the same `wp_` prefix.

A screenshot of the WordPress installer's database connection screen. The browser address bar shows 'wpjs.co/wp-admin/install.php'. The WordPress logo is centered at the top. Below it, a text box contains the instruction: 'Below you should enter your database connection details. If you're not sure about these, contact your host.' The form consists of five rows, each with a label, an input field, and a description: 'Database Name' with 'wordpress' in the field and 'The name of the database you want to use with WordPress.'; 'Username' with 'username' in the field and 'Your database username.'; 'Password' with 'password' in the field and 'Your database password.'; 'Database Host' with 'localhost' in the field and 'You should be able to get this info from your web host, if localhost doesn't work.'; and 'Table Prefix' with 'wp_' in the field and 'If you want to run multiple WordPress installations in a single database, change this.' A 'Submit' button is located at the bottom left of the form.

Once you click submit, WordPress will connect to your database. If the connection is successful, you'll be asked to set the site title and WordPress administrator credentials.



Once you've filled that in, click *Install WordPress*. The next screen should be the last one, congratulations!



Simply log in and enjoy your new WordPress site.

What if the installer doesn't work?

A few items to check for:

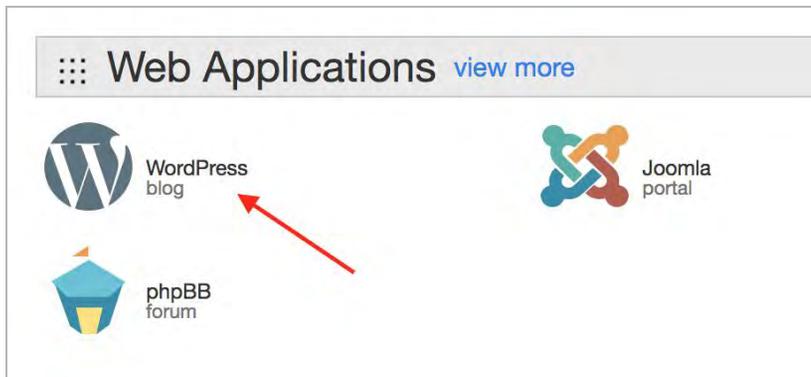
- Did you create and connect the database & database user?
- Did you use the right database credentials?
- Did all of the files get uploaded to the server?

The WordPress Codex provides further documentation on [troubleshooting common installation problems](#) and [installation FAQs](#).

Installing WordPress via cPanel

The manual installation is a great exercise for learning how WordPress works. But what if you need to get an installation done as quickly as possible?

Thankfully, cPanel has us covered there, too. From the cPanel main screen, under *Web Applications*, click *WordPress*.



On the WordPress screen, click *Install this application*.



You'll be walked through the installation process and asked a series of questions about your site: your domain name, your email address, plugins you want to use, and so on.

The default settings will usually suffice for most WordPress installations - review them, then click *Install*.

Once you click Install, a progress bar will appear, indicating the status of your WordPress installation.

When it finishes you'll see details about the site alongside links to tools and additional resources.



Final thoughts on installing WordPress with cPanel

We've looked at two different methods for installing WordPress using cPanel. The manual installation is a little complicated, but if you're new to WordPress, it's a helpful exercise. Understanding how WordPress works, and knowing how to access the files directly, is incredibly valuable for troubleshooting WordPress issues in the future.

Once you're comfortable installing WordPress manually, and simply need to get it done, the cPanel installer can save a lot of time.

Related reading

- [What to do after installing WordPress](#)
- [How to secure your WordPress site](#)
- [Hardening your WordPress installation](#)
- [How to enable Gzip compression on WordPress](#)

Managing WordPress with cPanel.

Managing WordPress with cPanel

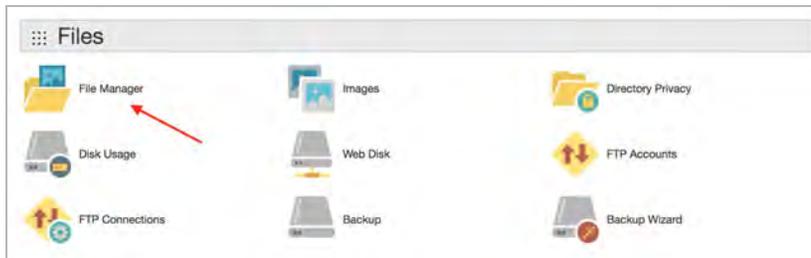
By Topher Derosia

cPanel is a handy tool for managing WordPress websites. Here are some of the great things you can do with it.

Manually upload plugins and themes

You can usually upload plugins and themes via the WordPress admin, but sometimes that doesn't work. The files may be too big, for example. Or there might be buggy code in a plugin or theme preventing you from accessing the WordPress admin area at all.

Fortunately, cPanel has an FTP client built in. From the cPanel main screen, under *Files*, click *File Manager*.



cPanel's File Manager allows us access the files on our hosting account without needing to use a standalone FTP client. To access WordPress plugins and themes, click on the `wp-content` folder.

Name	Size	Last Modified	Type
cg-bin	4 KB	Yesterday 9:40 AM	httpd
wp-admin	4 KB	Jun 8, 2017 10:29 AM	httpd
wp-content	4 KB	Today 6:54 PM	httpd
wp-includes	12 KB	Today 6:54 PM	httpd
404.shtml	1.66 KB	Dec 9, 2014 5:22 PM	text/h
home.html	1.92 KB	Dec 9, 2014 5:22 PM	text/h

From here, you can access either the plugins or themes directory, depending on what you're uploading.

In the example below, we're going to upload a plugin, so we click through to the plugins directory.

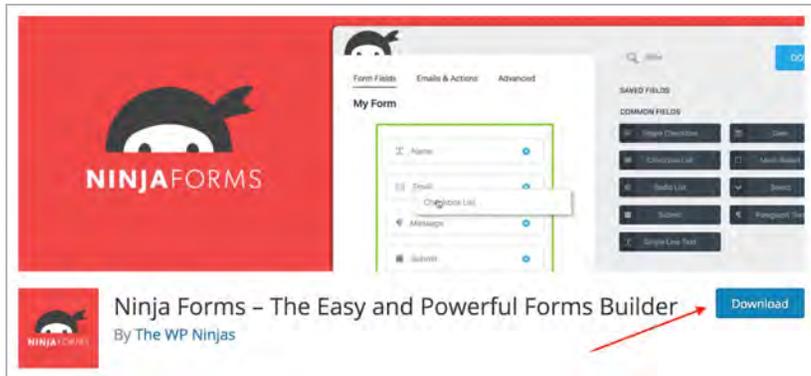
Name	Size	Last Modified
languages	4 KB	Today 6:54
plugins	4 KB	Today 6:54
themes	4 KB	Jun 8, 2017
uploads	4 KB	Today 6:54
index.php	28 bytes	Jan 8, 2012

Once we're in the plugins directory, we'll see a list of all the plugins currently installed on the site.

Name	Size	Type
akismet	4 KB	...
limit-login-attempts	4 KB	...
wp-easy-mode	4 KB	...
hello.php	2.2 KB	PHP
index.php	28 bytes	PHP

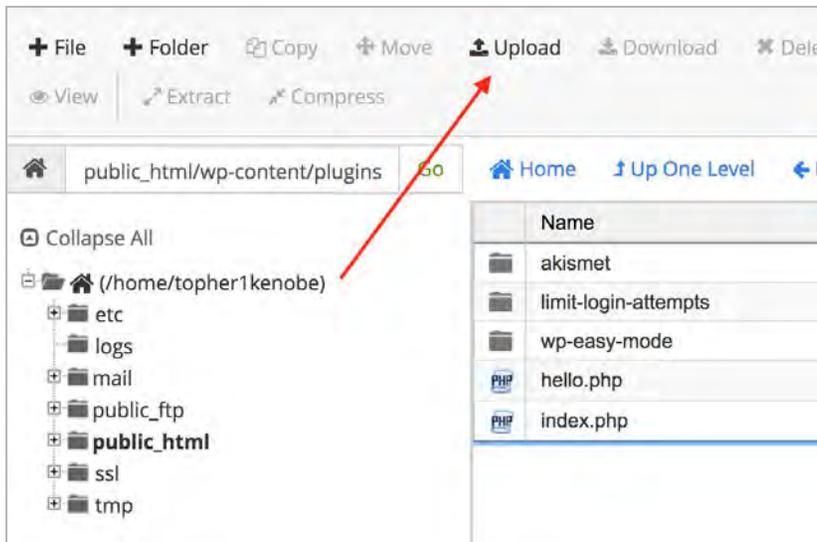
For this example I'm going to install a plugin called [Ninja Forms](#).

You can manually download plugins by going to their page on the WordPress Plugin Directory and clicking the *Download* button.



That will ask you to download a .zip file to your computer. (DO NOT unzip the file - we can unzip it on the server.)

In the File Manager, from the top toolbar, click the *Upload* button.

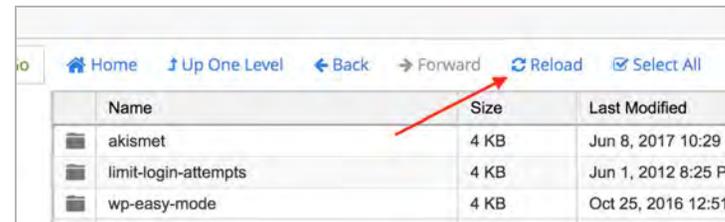


This will open a new tab (or window) asking you to set some permissions. Accept the default settings. Use the file chooser to select the plugin's .zip file that you just downloaded.

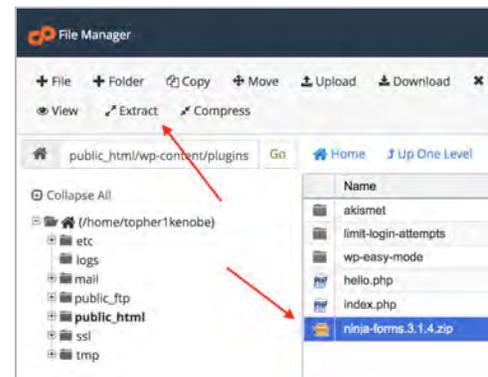
The File Manager will upload the .zip file to the server. As it's uploading you'll see an indicator in the bottom of the window, showing the progress of the upload.



You can safely close this window. In the File Manager toolbar, click the *Reload* button.



Your plugin .zip file should now be visible. Select it, and then from the toolbar, click *Extract*. The plugin is now installed and ready to be activated through the WordPress admin backend.



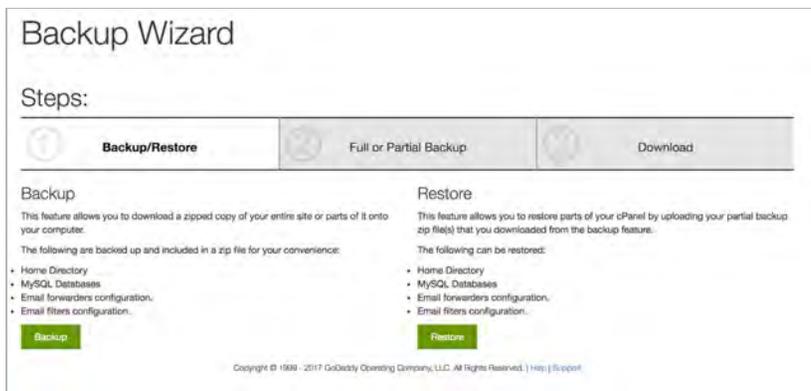
Running backups

You can also run backups in cPanel. From the cPanel main screen, under *Files*, there are two options: *Backup* and *Backup Wizard*.

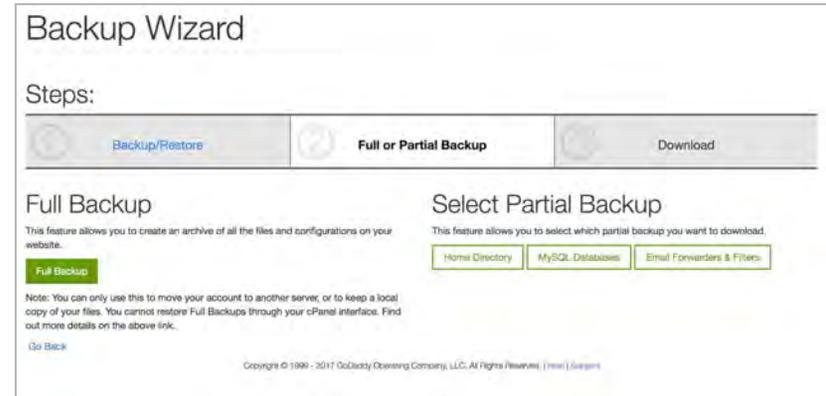


The *Backup* tool offers a few options to backup various parts of your site, and to restore previous backups. *Backup Wizard*, by comparison, walks you through some questions to determine what you want to back up.

The first screen simply asks if you want to back things up, or restore.



If you choose *Backup*, you'll be asked if you want to do a full backup, or if you want to do a partial backup of specific items.



Once you choose an option it moves to the *Download* section.



And on the *Download* section, under *Final Step*, just click to download the backup to your computer.

Restoring backups

Using the same Backup Wizard above, choose *Restore*. You'll be asked what kind of restore to perform.



The screenshot shows the 'Backup Wizard' interface. At the top, there are three steps: 'Backup/Restore', 'Select Restore Type', and 'Restore'. The 'Select Restore Type' step is active. Below the steps, there is a section titled 'Select Restore Type' with the text 'This feature allows you to select what to restore.' There are three buttons: 'Home Directory', 'MySQL Databases', and 'Email Forwarders & Filters'. A 'Go Back' link is at the bottom left. Copyright information for GoDaddy is at the bottom.

Under *Select Restore Type*, choose the option that matches your restore file. You'll be taken to the final Restore step.



The screenshot shows the 'Backup Wizard' interface at the 'Final Step'. The steps are 'Backup/Restore', 'Select Restore Type', and 'Restore'. The 'Restore' step is active. Below the steps, there is a section titled 'Final Step' with the text 'Restore your backup.' and 'Restore a Home Directory Backup'. There is a 'Choose file' section with 'No file chosen.' and an 'Upload' button. A 'Go Back' link is at the bottom left. Copyright information for GoDaddy is at the bottom.

Select the backup file on your computer, then click *Upload*. The Backup Wizard will complete the restore process for you.

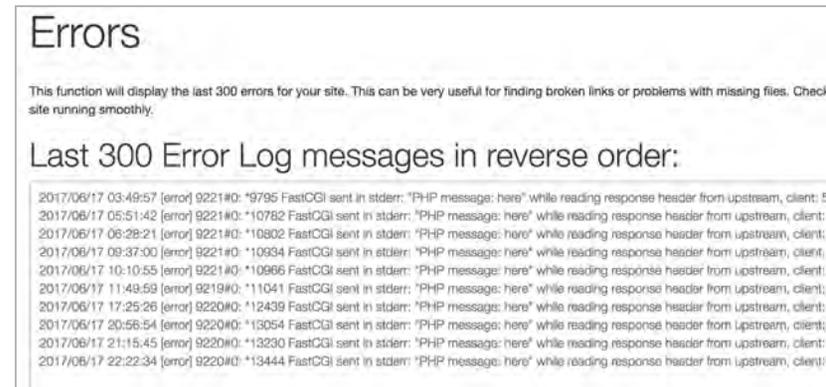
Accessing error logs

Sometimes things won't work on a site and there's no indication of what's going wrong. When this happens it can be handy to look at your server's error logs.

In cPanel, under *Metrics*, click *Errors*.



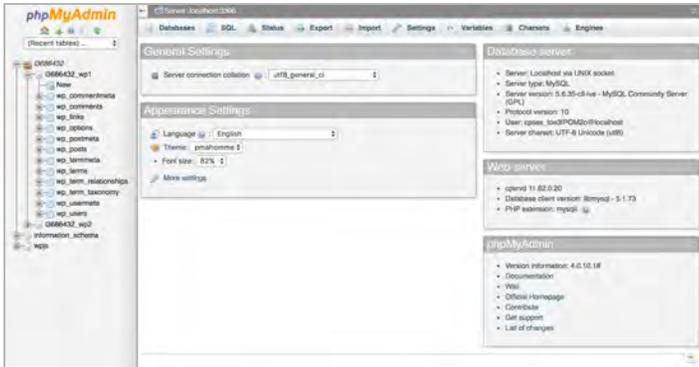
On the *Errors* screen, the last 300 error log messages will be visible.



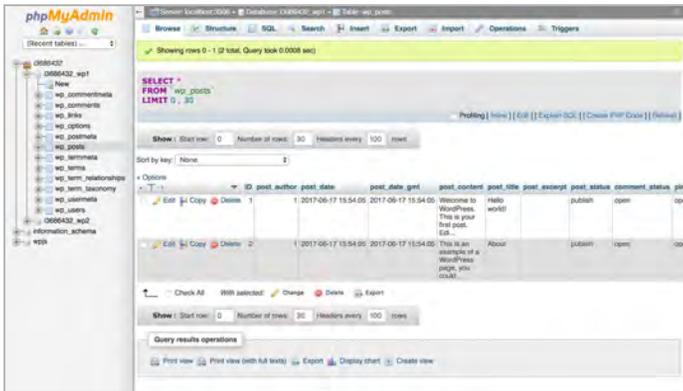
Related reading:

- [Setup PHP error logging](#)
- [Set up custom-defined PHP error logs in cPanel](#)
- [Working with cPanel logs & metrics](#)
- [Nine best practices for website backups](#)

Expand any of these databases and you should see tables that match the WordPress prefix you set when you installed (e.g. wp_).



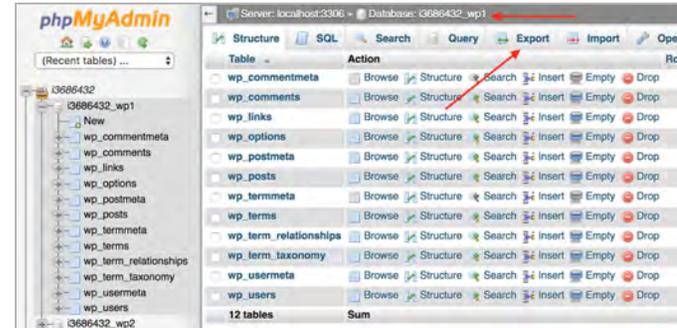
Click any one of those tables you'll be able to browse the data stored within. In the screenshot below I chose wp_posts, the table where all WordPress post content is stored.



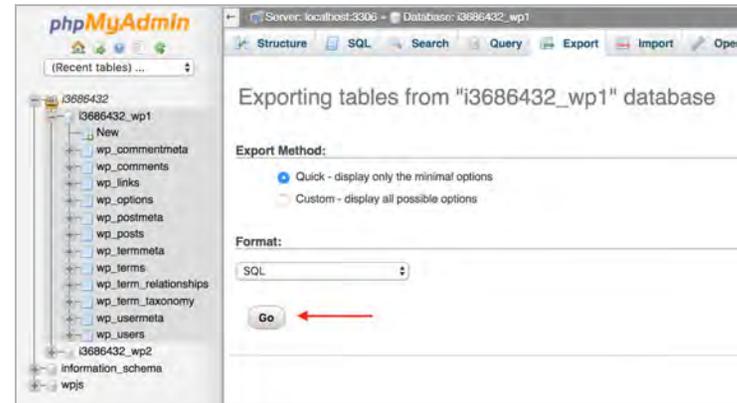
Clicking Edit will let you modify the data in that row. Keep in mind that much of the data in WordPress is spread across more than one table, so be very careful when editing raw data.

Exporting your database with phpMyAdmin

From the main database screen you'll see a list of all the tables present in the database. To export everything, click *Export*.



On the next screen, you can simply click the Go button and you'll be asked to save a .sql file to your computer.



Now you have a .sql backup of your database.

Troubleshooting with WP_DEBUG

You can make WordPress print errors right to the browser. This usually isn't a good idea on a publicly-accessible site, but sometimes it's necessary, and it can be very handy on a development or staging site. Here's how you do it.

In your main WordPress folder, there's a file called `wp-config.php`. Inside there you'll find WordPress settings like your username, password, etc. There's also a setting called `WP_DEBUG`.

```
80 /**
81  * For developers: WordPress debugging mode.
82  *
83  * Change this to true to enable the display of notices during development.
84  * It is strongly recommended that plugin and theme developers use WP_DEBUG
85  * in their development environments.
86  *
87  * For information on other constants that can be used for debugging,
88  * visit the Codex.
89  *
90  * @link https://codex.wordpress.org/Debugging_in_WordPress
91  */
92 define('WP_DEBUG', false);
```

Change

```
define('WP_DEBUG', false);
```

to:

```
define('WP_DEBUG', true);
```

and WordPress will start printing errors to the browser.

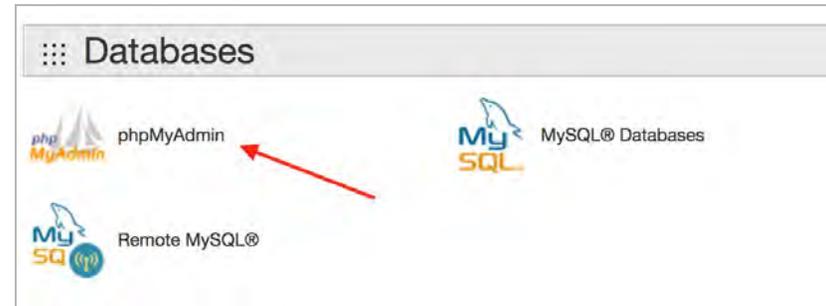
Keep in mind that there can be many errors and warnings that do NOT break your site, are usually hidden by `WP_DEBUG` being disabled. Warnings don't necessarily mean your site is broken.

Related: [Suppressing warning messages in PHP](#)

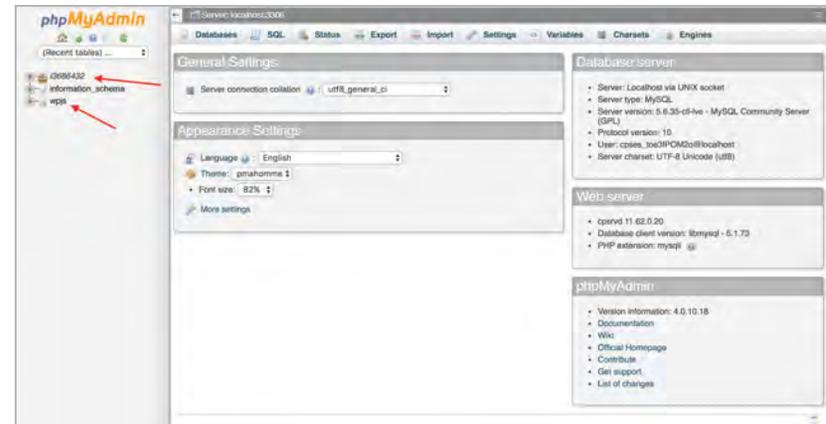
Using phpMyAdmin with your database

WordPress stores its data in a MySQL database. To access the database directly, we can use a tool called phpMyAdmin.

From the cPanel main screen, under *Databases*, click *phpMyAdmin*.



You'll be taken to a new tab with the phpMyAdmin interface. On the left sidebar you'll see a list of databases present on the server.

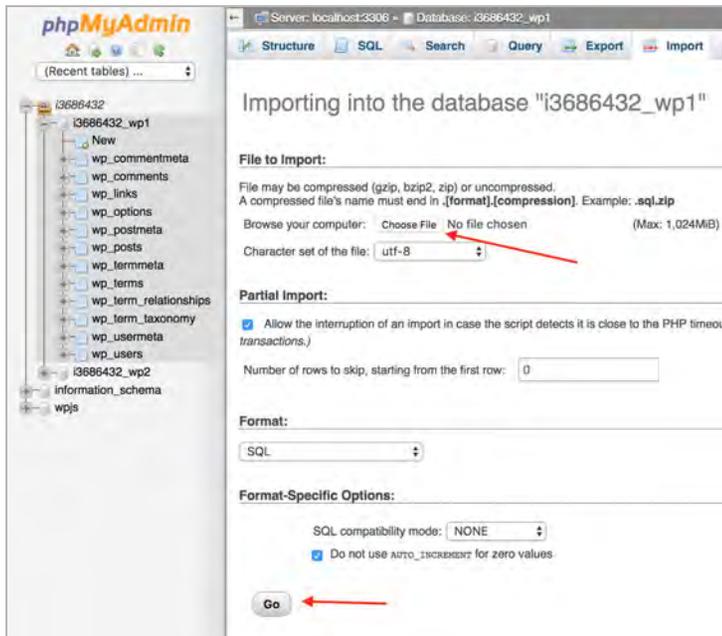


Importing a database SQL file with phpMyAdmin

In phpMyAdmin, on the main screen for a database next to Export, click *Import*.



Select the database .sql file from your computer that you want to import. Keep the default settings. Click Go and the database will be imported.



Final thoughts on managing WordPress with cPanel

There are many more things that can be done with cPanel to make WordPress management a breeze. I highly recommend you read through each feature and see what it can do for you.

Related reading

- [How to migrate a WordPress site to a new host](#)
- [Creating MySQL databases in cPanel](#)
- [Connecting to MySQL using PHP](#)
- [Official phpMyAdmin documentation](#)

Advanced cPanel features for WordPress

Advanced cPanel features for WordPress

By Saurabh Shukla

With cPanel you can manage many aspects of your server without much complexity. For instance, you can easily set up domains, databases, and email accounts in a few clicks.

But did you know that with cPanel, you can also block malicious bots and hackers, prevent bandwidth abuse, and make parts (or the whole) of your WordPress website private?

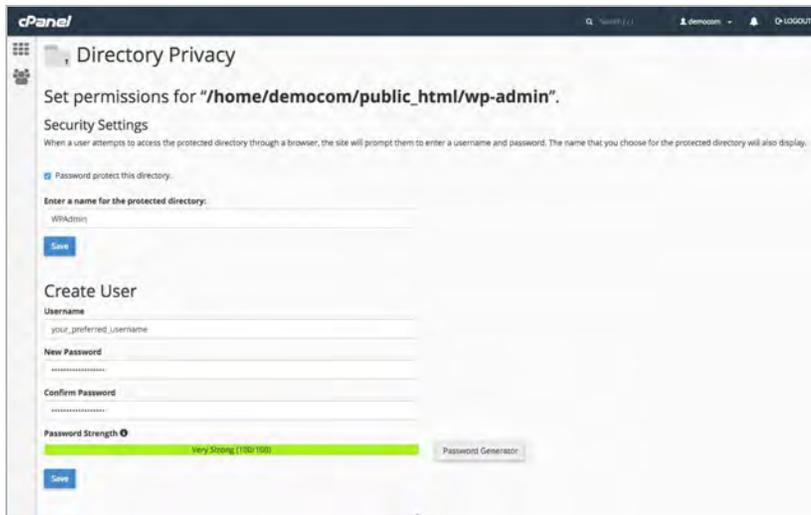
Another feature of cPanel is SSH, or Secure Shell access, which allows you to log into the server remotely and run commands that let you perform admin tasks more efficiently.

In this section of our guide we'll cover some of these cPanel features, plus even more advanced capabilities that require some familiarity with basic shell commands.

WordPress privacy and security with cPanel

Directory privacy & security

You can make a directory on your site private by adding a username or password to it. For example, you can make your entire site private by protecting the main directory that contains your WordPress site.



Anyone who tries to open the website in their browser will be prompted for credentials.



What can you do with directory privacy? A few ideas:

Private before launch: This could be really useful when you are still building the site. By making your WordPress directory private, you can hide the site from general public until everything is ready.

Staging site: Create a “staging” copy of your site to test changes before you apply them to your live site, e.g. on a subdomain like `staging.yourdomain.com`. Password protect the parent directory of the staging site so it's not accessible by strangers or search engines.

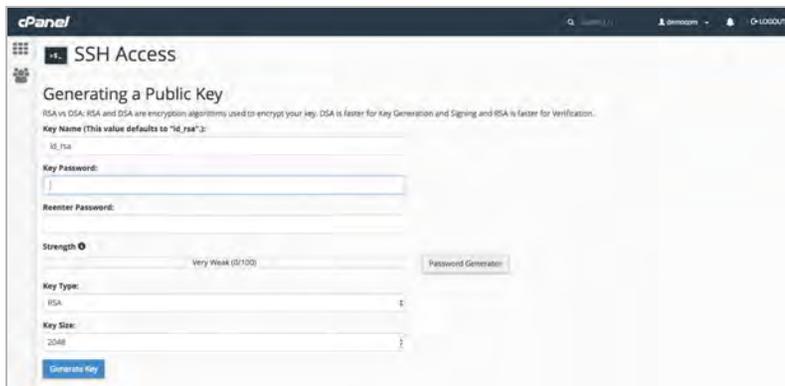
Extra security: You could also make the `/wp-admin/` directory private. This way, you get an additional level of security beyond the WordPress login form. A potential hacker will have to go through an extra step to access your admin area.

Running commands remotely (SSH access)

The SSH Access feature allows you to remotely work on the server from the **terminal** on your Mac or Linux machine. On Windows, you can use [bash](#) (as of Windows 10) or a software called [PuTTY](#) to do the same on older versions of Windows.

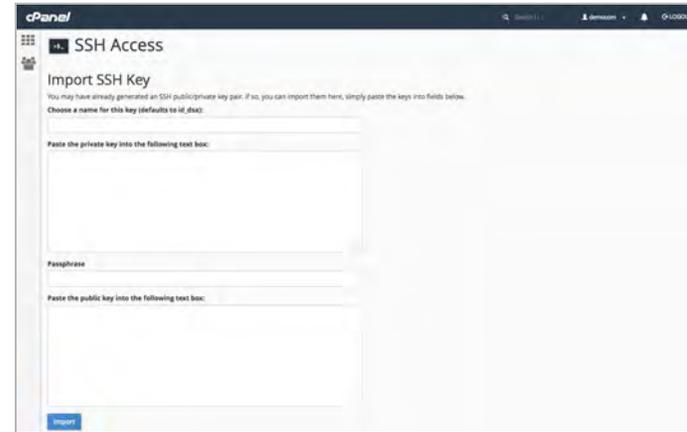
You need a pair of public and private SSH keys to use SSH Access. You can generate the public-private key pair on the server and download and configure the keys on your machine.

To enable SSH: From the cPanel main screen, under *Security*, click *SSH Access*.



The screenshot shows the 'Generating a Public Key' form in cPanel. It includes a 'Key Name' field with a default value of 'id_rsa', a 'Key Password' field, a 'Reenter Password' field, a 'Strength' dropdown set to 'Very Weak (51200)', a 'Key Type' dropdown set to 'RSA', and a 'Key Size' dropdown set to '2048'. A 'Generate Key' button is at the bottom left.

If you already have such keys on your machine, you can just import them into cPanel:



The screenshot shows the 'Import SSH Key' form in cPanel. It includes a 'Choose a name for this key (defaults to id_rsa)' field, a 'Paste the private key into the following text box:' field, a 'Passphrase' field, and a 'Paste the public key into the following text box:' field. An 'Import' button is at the bottom left.

Once you've set up the keys, you can open a terminal on Mac or Linux (or PuTTY on Windows) on your local machine and run the following command to remotely login:

```
ssh -p 22 cpanel_username@yoursite.com
```

After you're logged in, it'll be like running commands directly on your server's shell. (SSH Access = Secure Shell Access).

The commands that you're allowed to run will vary from one hosting provider to another. Yet, the basic commands will be the same across hosts. You can perform a large number of file and directory operations. No need to use an FTP software (FileZilla) or login into cPanel.

Basic shell commands

Combining basic shell commands with cPanel can make your life even easier. It lets you schedule repetitive tasks using Cron Jobs.

For example, you could set a cron job to run commands to take a backup of your files and database, once every week. It could save you a lot of time if you do such things manually.

Related: [How to use WP-CLI](#)

I strongly recommend learning shell commands if you want to level up as a WordPress developer. For example, consider the following command:

```
mkdir /var/www/new_directory
```

What do you think it does? Would it surprise you to know that it makes a new directory called `new_directory` inside the `www` folder?

Consider another example:

```
mv /var/www/old_directory /var/www/new_directory
```

You're right if you guessed that this command moves everything from `old_directory` to `new_directory`.

There are many more commands that you can do much more with. Here's a reference to [basic shell commands](#) to get you started. The best part is, you don't even have to remember any of those commands. Just go back to the reference or simply Google it.

To set up shell access: From the main cPanel screen, under *Security*, click *SSH Access* and follow the on-screen instructions.

cPanel Cron Jobs and WordPress

You might be familiar with [Scheduled Posts](#) in WordPress where you can set a post to publish at a particular time and date. (Some plugins also schedule tasks like regular automated backups.)

These scheduled tasks are run by [WP-Cron](#). Unfortunately, WP-Cron can fail to run for a number reasons.

A more dependable way to schedule such tasks is to use Cron Jobs that are run by Linux Cron. Let's try and understand the difference.

WP-Cron vs Linux Cron

A scheduled task with WP-Cron, like publishing a post or saving a backup, will only run if WordPress is loaded. But WordPress only loads when someone visits the site (front-end or back-end), at or after the scheduled time. If no one visits the site on the schedule, WordPress is never loaded, and the cron job is never run.

That's why WP-Cron may fail because of a large server load, DNS-related issues, or caching. At times, bugs in WordPress or plugin conflicts can also cause issues. That's why it's called a *pseudo* cron.

Cron Jobs set through cPanel are run by your server's Linux Cron and don't rely on WordPress being loaded by a user. It's a *true* cron.

Switching WP-Cron to cPanel cron jobs

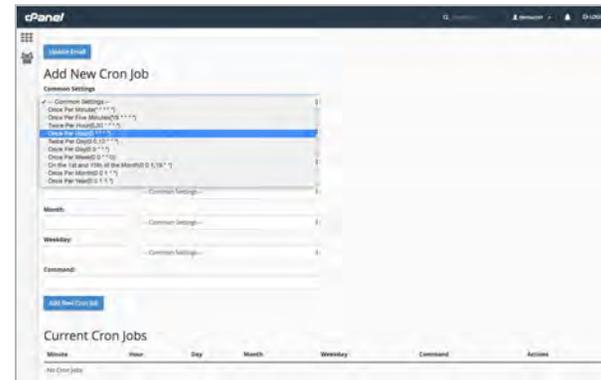
Disabling WP-Cron: Open `wp-config.php` (located in the directory where WordPress is installed). Create a new line after `<?php`, and add the following code on the new line:

```
define( 'DISABLE_WP_CRON' , true );
```

Switching to cron jobs

From the cPanel main screen, under *Advanced*, click *Cron Jobs*.

On the Cron Jobs screen, under *Add New Cron Job*, select *Once per hour* or *Once per day* to set the frequency. You can adjust the timing and schedule as per your needs. (For example, if posts are scheduled to publish at 10:00 am everyday, set the Cron Job to run at that time.)



Add the following in the command input (replace `yoursite.com` with the actual domain name of your site):

```
wget -q -O -  
http://yoursite.com/wp-cron.php?doing_wp_cron >/dev/null  
2>&1
```

So what's happening here? The Linux Cron command will load the WordPress file that runs all the cron jobs set in WordPress, and ensures that all the tasks are completed without any delays or issues.

Scheduling backups

Most hosting providers, including GoDaddy, create a frequent automated backup of your site. But you can also schedule a backup of your database and WordPress files yourself - you just need to add relevant commands to new cron jobs.

If you're not comfortable with writing commands, you can find WordPress plugins to perform automated scheduled backups. Then after switching from WP-Cron to Cron Jobs, your backups will happen automatically (depending on the plugin's settings, of course).

Final thoughts on advanced cPanel features

cPanel has a lot of useful features for WordPress developers. I have explored just a few of them. Ideally, I'd recommend exploring each feature and reading [relevant documentation provided by your web host](#). With the right understanding of cPanel features, you can level up and learn to tweak your WordPress site in powerful ways.

Next Steps

Ready to start using cPanel and WordPress together? [GoDaddy's new Business Hosting plans](#) combine the power and freedom of VPS with the easy-to-use convenience of cPanel. It's ideal for hosting multiple WordPress installations on the same account, or for hosting more resource-intensive websites like eCommerce.

[» Learn more about Business Hosting from GoDaddy](#)

The end.

