

# WP-CLI

WP-CLI is a command-line interface for WordPress that you can access when [connected via SSH](#). This guide covers the most commonly used commands and their associated sub-commands.

This feature is available on sites with the [WordPress.com Creator or Entrepreneur plan](#).

## Access CLI

[Top ↑](#)

WP-CLI comes preinstalled on WordPress.com and extends the shell to provide WordPress-specific command line tools. You can start running WP-CLI commands once you have connected to SSH. Visit the [SSH guide to learn how to activate and use SSH](#).

## Common Commands

[Top ↑](#)

There are a number of commands that you may find helpful when working with your site. These commands and their common parameters are included below. There are also many other useful commands available, and these sections do not include them all. Please refer to the [WordPress.org Developer CLI Commands Guide](#) for more information.

In the CLI itself you can also make use of the `help` command to find more information about what parameters and flags are available. For example: `wp help plugin`.

## Manage Users

Use the following commands to manage your site’s local users. Please keep in mind that changing user aspects in CLI will not change the settings for WordPress.com user accounts that may be associated with local accounts.

The user serving as your primary Jetpack connection must have an email that matches the email of the **site owner’s** WordPress.com account. If the email does not match, the site’s Jetpack Connection will not work correctly. The username does not need to match.

Command	Description
<code>wp user list</code>	List all users on the site
	If you have many users you may wish to restrict

Command	Description
	the list to specific roles by adding <code>--role={ROLE}</code>
<code>wp user get {USER ID/NAME/EMAIL}</code>	Return a specific user
<code>wp user create {NAME} {USER EMAIL} --role={ROLE}</code>	<p>Create a new user with the name, user email, and role defined</p> <p>This command will automatically create and return a secure password for the user</p> <p>This command does not create a matching WordPress.com user. If you are using <a href="#">WordPress.com Secure Sign-On (SSO)</a>, you should instead create the local user by inviting a user via <i>Users &gt; Add New</i>. This will ensure a WordPress.com account and local user are both created.</p>
<code>wp user update {USER ID/NAME/EMAIL} -{-{KEY}={VALUE}}</code>	<p>Update a user</p> <p>Keys include <code>display_name</code>, <code>user_email</code>, <code>role</code>, and <code>user_pass</code></p>
<code>wp user reset-password {USER ID/NAME/EMAIL}</code>	<p>Reset a user password with a random secure string</p> <p>This command can take multiple users if you put a space between each ID</p> <p>If you do not wish to notify users of password resets you can add <code>--skip-email</code></p>
<code>wp user add-role {USER ID} {ROLE}</code>	<p>Add a role to a user</p> <p>This can be used to add default and custom roles</p>
<code>wp user remove-role {USER ID} {ROLE}</code>	Remove a role from a user
<code>wp user delete {USER ID/NAME/EMAIL} -{-reassign={USER ID/NAME/EMAIL}}</code>	<p>Delete a user</p> <p>If you do not include the <code>--reassign={USER ID/NAME/EMAIL}</code> flag all content created by the user will be deleted</p>

**Note:** You will not be able to update a user's `user_login` with the `wp user update` command. Instead, updating the `user_login` requires using `wp search-replace`. This change, however, is not recommended as it performs partial replacements. For example: If there are two users with the usernames "techie" and "nontechie", replacing "techie" with "sense" will result in "sense" and "nonsense".

If you would still like to replace a `user_login` the command to use is:

```
wp --skip-plugins --skip-themes search-replace "old username" "new username"
*users --include-columns=user_login,user_nicename --dry-run
```

## Manage Plugins and Themes

Use the following commands to manage your plugins and themes. When working with plugins and themes, please note that the related slug will not always match the plugin/theme name.

If you are having issues with a plugin or theme you can start your command with `wp --skip-themes --skip-plugins` to run the commands despite conflicts.

You can find [more information about skip commands below](#).

Command	Description
<code>wp plugin list</code> <code>wp theme list</code>	List all the plugins/themes installed on the site and their status. From these lists you will find the slugs of your installed plugins and themes, which you can use for the purpose of running the commands below.
<code>wp plugin activate {SLUG}</code> <code>wp theme activate {SLUG}</code>	Activate the specified plugin/theme  To deactivate a theme you need to activate another in its place
<code>wp plugin deactivate {SLUG}</code>	Deactivate the specified plugin
<code>wp plugin delete {SLUG}</code> <code>wp theme delete {SLUG}</code>	Delete the specified plugin/theme  You cannot delete an active theme or managed themes with this command. To remove an active theme, switch to a different theme first, then delete the theme that you want. To

Command	Description
	remove managed themes, <a href="#">you may do so via SFTP</a> .
<code>wp plugin install {PLUGIN SLUG or URL TO INSTALLATION_FILE}</code>	Install a plugin from the WordPress.org repo based on the plugin's slug or via a URL path to an installation .zip file
<code>wp theme install {PLUGIN SLUG or URL TO INSTALLATION_FILE}</code>	<p>You can not overwrite an existing plugin/theme unless you run the command with the <code>--force</code> flag. You can use this to update plugins and fix corrupt plugin folders.</p> <p>To automatically activate the plugin after installation, include the <code>--activate</code> flag</p>
<code>wp plugin update {SLUG}</code> <code>wp theme update {SLUG}</code>	<p>Update the specified plugin if available. You cannot use this command to update managed plugins/themes.</p> <p>You can also use the <code>--all</code> flag if you want to update all plugins</p>
<code>wp plugin auto-updates {COMMAND} {PLUGIN SLUG}</code>	<p>Display the status of plugin updates or enable/disable updates for a specific plugin</p> <p>COMMANDS are <code>status</code>, <code>enable</code>, <code>disable</code></p> <p>You cannot disable updates for managed plugins or themes</p>

## Manage Jetpack

Use the following commands to adjust Jetpack modules, check your Jetpack connection, or troubleshoot Jetpack issues.

Disconnecting Jetpack will disable many important WordPress.com features including general site and billing management.

Using `--skip-plugins` will prevent these commands from working.

You can find [more information about skip commands below](#).

Command	Description
<code>wp jetpack module list</code>	List the Jetpack modules and their statuses
<code>wp jetpack module activate {MODULE NAME}</code>	Activate the specified Jetpack module  You can also use <code>all</code> in place of <code>{MODULE NAME}</code>
<code>wp jetpack module deactivate {MODULE NAME}</code>	Deactivate the specified Jetpack module  You can also use <code>all</code> in place of <code>{MODULE NAME}</code>
<code>wp jetpack status</code>	Report the status of Jetpack, including which user account is the primary connection (this should always be the site owner)  An additional parameter, <code>full</code> , can be added to report full details, including Jetpack version, WordPress version, and module status
<code>wp jetpack sync start</code>	Force a new Jetpack sync, between WordPress.com and the WordPress installation  <b>Note:</b> Jetpack automatically syncs at certain intervals so this command is typically not necessary. If you do run this command, be advised that it is resource intensive and can slow your site until the sync is complete
<code>wp jetpack disconnect blog</code>	Disconnect the currently connected site  If you disconnect Jetpack you will need to

Command	Description
	reconnect it in order for your site to function correctly. A disconnected Jetpack will prevent your site from properly backing up data and will interfere with Jetpack functions such as scheduled posts, subscriptions, VideoPress, etc.
<code>wp jetpack disconnect user {USER ID/NAME/EMAIL}</code>	Disconnect a specific user from the currently connected site

While your [site's visibility is set to Private](#), some Jetpack modules are disabled and cannot be managed in CLI. These modules are: Ads, Asset CDN, Enhanced Distribution, Google Analytics, Image CDN, JSON API, Publicize, Sharing, Site Verification, and Sitemaps.

## Manage Site Settings

Use the following commands to manage various aspects of your site and users.

Command	Description
<code>wp site empty --yes</code>	Empty a site of content (posts and pages)
<code>wp site empty --uploads --yes</code>	Empty the site, including media files

`wp site empty` commands are irreversible. If you use them, your content will be removed permanently. To recover them, [you will need to restore from a previous backup](#). Please ensure you have a backup available before using the function.

`wp site` commands will not delete themes or plugins, nor will they reset a theme or erase plugin settings. If you need to reset your site completely, [please reach out to WordPress.com support](#).

Command	Description
<code>wp option get {OPTION}</code>	Return the value of an option
<code>wp option update {OPTION} {VALUE}</code>	Update the value of an option
<code>wp option add {OPTION} {VALUE}</code>	Add a new option
<code>wp option delete {OPTION}</code>	Delete an option

The options table is the “control center” of a website. If an option is improperly edited or deleted, you may stop your entire site from working.

When working with `wp option` commands, if you get the following error in CLI it may mean the options table has been corrupted:

One or more database tables are unavailable. The database may need to be repaired.

You can restore the table by rewinding the site’s SQL table to a last good copy. Please note, this will reverse all mutable changes to the database that occur after the restore point and can result in lost data.

To restore an SQL table:

1. In the site’s dashboard, navigate to *Jetpack > Activity Log*.
2. Select a [backup point](#) to rewind to.
3. Uncheck all boxes except for `Site Database (SQL)`.
4. Click **Confirm Rewind**.

Command	Description
<code>wp role list</code>	Return a list of roles

Command	Description
<code>wp role reset {ROLE}</code>	<p>Reset the specified role</p> <p>You can list multiple roles (without comma)</p> <p>You can also replace {ROLE} with <code>--all</code> to reset all at once</p>

## Manage Content

You can use the following commands to manage your site content, including posts/pages, menus, and media.

Command	Description
<code>wp post list</code>	<p>Return a list of blog posts</p> <p>You can use a variety of flags with this command, including:</p> <ul style="list-style-type: none"><li><code>--format={FORMAT}</code></li><li><code>--posts_per_page={#}</code></li><li><code>--post_type={TYPE}</code></li></ul> <p>Post types include <code>posts</code>, <code>pages</code>, <code>attachments</code>, and <code>products</code></p> <p>Multiple flags can be used at once</p> <p>ex: <code>wp post list --post_type=posts,pages --format=count</code></p>
<code>wp post update {POST_ID} --{KEY}={VALUE}</code>	<p>Update post information</p> <p>ex: <code>wp post update 123 --post_status=draft</code></p>
<code>wp post delete {POST_ID}</code>	<p>Delete a post</p>
<code>wp post exists {POST_ID}</code>	<p>Check if a post exists</p>



Command	Description
<code>wp menu list</code>	List menus on the site
<code>wp menu item list {TERM_ID}</code>	List menu items of a specific menu
<code>wp menu item update {MENU_ITEM_PARENT_ID}</code>	Update a specific menu item

The `wp menu` commands can be useful if you see the primary menu freezing in the Customizer or `/wp-admin` screen.

Command	Description
<code>wp media regenerate</code>	Regenerate media thumbnails when the Jetpack Photon module is deactivated  You can add <code>--only-missing</code> to the end to skip existing image sizes

## Troubleshooting Site Issues

You can use the following commands when troubleshooting issues on your site. Checking for PHP errors can help you identify sources of errors on your site or use `wp wpcomsh` commands to quickly perform plugin conflict checks or to adjust whether or not you are using managed versions of plugins.

Both Jetpack and Akismet are managed plugins that are required for your site to work correctly here at WordPress.com. These plugins should remain activated and cannot be switched to unmanaged versions.

Command	Description
<code>wp php-errors</code>	Output error logs in WP-CLI  This command only outputs PHP errors, not of any other language such as HTML, Javascript, or CSS. To see these other errors, use the Browser Inspector  You can add <code>--limit={#}</code> to reduce

Command	Description
	the number of errors shown to improve readability
<code>wp wpcomsh deactivate-user-plugins</code>	<p>Deactivate user plugins</p> <p>You can add the <code>--interactive</code> flag to the end of the command to force confirmation for each plugin</p> <p>This command will not deactivate Akismet, Jetpack, or managed ecommerce plugins</p>
<code>wp wpcomsh reactivate-user-plugins</code>	<p>Reactivate user plugins</p> <p>You can add the <code>--interactive</code> flag to the end of the command to force confirmation for each plugin</p>
<code>wp wpcomsh persistent-data {DATA}</code>	<p>Return persistent data (which Jetpack uses to verify site capabilities)</p> <p>Available <code>{DATA}</code> include:</p> <ul style="list-style-type: none"> <li><code>WPCOM_PLAN_EXPIRATION</code></li> <li><code>JETPACK_BLOG_TOKEN</code></li> <li><code>WPCOM_PURCHASES</code></li> <li><code>WPCOM_PUBLIC_COMING_SOON</code></li> </ul>
<code>wp wpcomsh plugin use-managed {SLUG}</code>	<p>Use a managed (<a href="#">symlinked</a>) version of a plugin</p> <p>This command will remove existing, unmanaged versions and replace them with managed versions. It is available for <code>amp</code>, <code>classic-editor</code>, <code>coblocks</code>, <code>crowdsignal-forms</code>, <code>full-site-editing</code>, <code>gutenberg</code>, <code>layout-grid</code>, <code>page-optimize</code>, <code>wp-seo</code>, <code>woocommerce</code>, WooCommerce Extensions from the WordPress.com Marketplace and <a href="#">plugins included in the Entrepreneur plan</a>.</p> <p>You can add <code>--remove-existing</code> to skip the confirmation step</p>
<code>wp wpcomsh theme use-managed {SLUG}</code>	<p>Use a managed (<a href="#">symlinked</a>) version of a theme</p> <p>This command will remove existing, unmanaged versions and replace them with managed versions. It is available for all WordPress.com themes,</p>

Command	Description
	Storefront, and any Twenty * themes (ex: Twenty Twenty Two)
	You can add <code>--remove-existing</code> to skip the confirmation step
<code>wp wpcomsh plugin use-unmanaged {SLUG}</code>	<p>Use an unmanaged (not <a href="#">symlinked</a>) version of a plugin</p> <p>This command will remove existing, managed versions and replace them with unmanaged versions from the WordPress.org plugin repository. It will not work for WooCommerce Extensions from the WordPress.com Marketplace and <a href="#">plugins included in the Entrepreneur plan</a>. You will need to install these plugins manually to get the unmanaged version.</p> <p>You can add <code>--remove-existing</code> to skip the confirmation step and <code>--version={X.Y.Z}</code> to define the version to install</p>
<code>wp wpcomsh theme use- unmanaged {SLUG}</code>	<p>Use an unmanaged (not <a href="#">symlinked</a>) version of a theme</p> <p>This command will remove existing, managed versions and replace them with unmanaged versions from the WordPress.org theme repository if available.</p> <p>You can add <code>--remove-existing</code> to skip the confirmation step and <code>--version={X.Y.Z}</code> to define the version to install</p>

The WordPress.com Site Helper or `wpcomsh` must-use plugin runs on sites with activated hosting features to make them work seamlessly in the WordPress.com ecosystem.

Command	Description
<code>wp cache flush</code>	<p>Flush the cache</p> <p>This command is not meant to be a fix for issues and can obscure root causes. Please use with care. Clearing the cache may also make your site run</p>

Command	Description
	slower while the cache is rebuilt on page load.

## ipping Plugins and Themes

You may want to skip plugins, and sometimes themes, when running CLI commands. This can be useful for troubleshooting conflicts or for overriding restrictions placed by plugins, such as control over roles.

`--skip-plugins` can be added before any command and skips plugins

`--skip-themes` can be added before any command and skips themes

You can use these commands together. You can also use these commands with slugs to skip specific plugins or themes. This can be useful for ruling out the effects of certain plugins or themes when deactivation is not possible.

For example: `wp --skip-plugins=woocommerce, code-snippets, elementor-pro plugin list`

## LI Restrictions

[Top ↑](#)

In order to provide a secure and performant environment, we restrict certain CLI commands. If a command is restricted, it will return `forbidden` in the CLI.

Typically, you can do pretty much anything on CLI that you could do with the GUI (Graphical User Interface) in your site's dashboard, but without any of the warnings or checks provided in the GUI. For example, the `wp site empty --yes` command will empty your site's content without confirmation. With these restrictions, it's harder to accidentally destroy a site.

## troubleshooting CLI

[Top ↑](#)

Occasionally, you may see HTML returned when you enter a command in the CLI. This can happen when you have a maintenance plugin active or can be a sign that a theme or plugin is coded incorrectly.

If no maintenance plugin is present, you can use the `--skip-*` commands to verify whether the source is a plugin or theme. Once identified, you can deactivate the plugin or theme and may want to reach out to the plugin developer to get it corrected.

Due to the complex nature of WP-CLI commands, we are not able to provide extensive support for using these tools. [Happiness Engineers are available to help with issues accessing WP-CLI](#) but cannot guide you through using commands.

---

## Additional WP-CLI Resources

[Top ↑](#)

[WordPress.org WP-CLI documentation](#)

[WooCommerce WP-CLI documentation](#)

[WP-CLI.org](#)