Taking A Deeper Look at Local by Flywheel

It is easy to think of Local by Flywheel as just a way to run a WordPress site locally. It is that and it is also much more than that.

Review: What is Local by Flywheel?

Local by Flywheel (Local) is a WordPress specific local development environment. It provides several features that make development easier and faster.

> Some of these features include:

- Multiple hosting options
    - Ngnix and Apache servers
    - Multiple versions of PHP (5.6, 7.3, 7.4, and 8.0) and MySQL (5.7 and 8.0)
    - The ability to swap between versions of PHP and MySQL
    - The ability to switch between Nginx and Apache
- Import and Export sites
    - Export and Import all relevant files from a WordPress site, including the database
- Create blueprints, templates for new sites
- Share your site during development
    - Ngrok tunneling
    - Live Links provide an ehanced tool to share your development site with other
- Headless WordPress
    - Create headless-optimized WordPress sites (Available in a plugin)
    - Automatically create and configure a Node.js frontend connected to a WordPress backend
- Extensibility via Local plugins

This post will concentrate on the command line features available in Local.

Additional Features: SSH, CLI and

associated tools

There was something I didn’t notice or understand about Local: how to use SSH and the command line tools in a Local environment.

SSH Access

To acccess the features we’ll discuss below, follow these steps:

1. Start the site you want to work with
2. Right Click the name of the site
3. Select ***Open Site Shell***

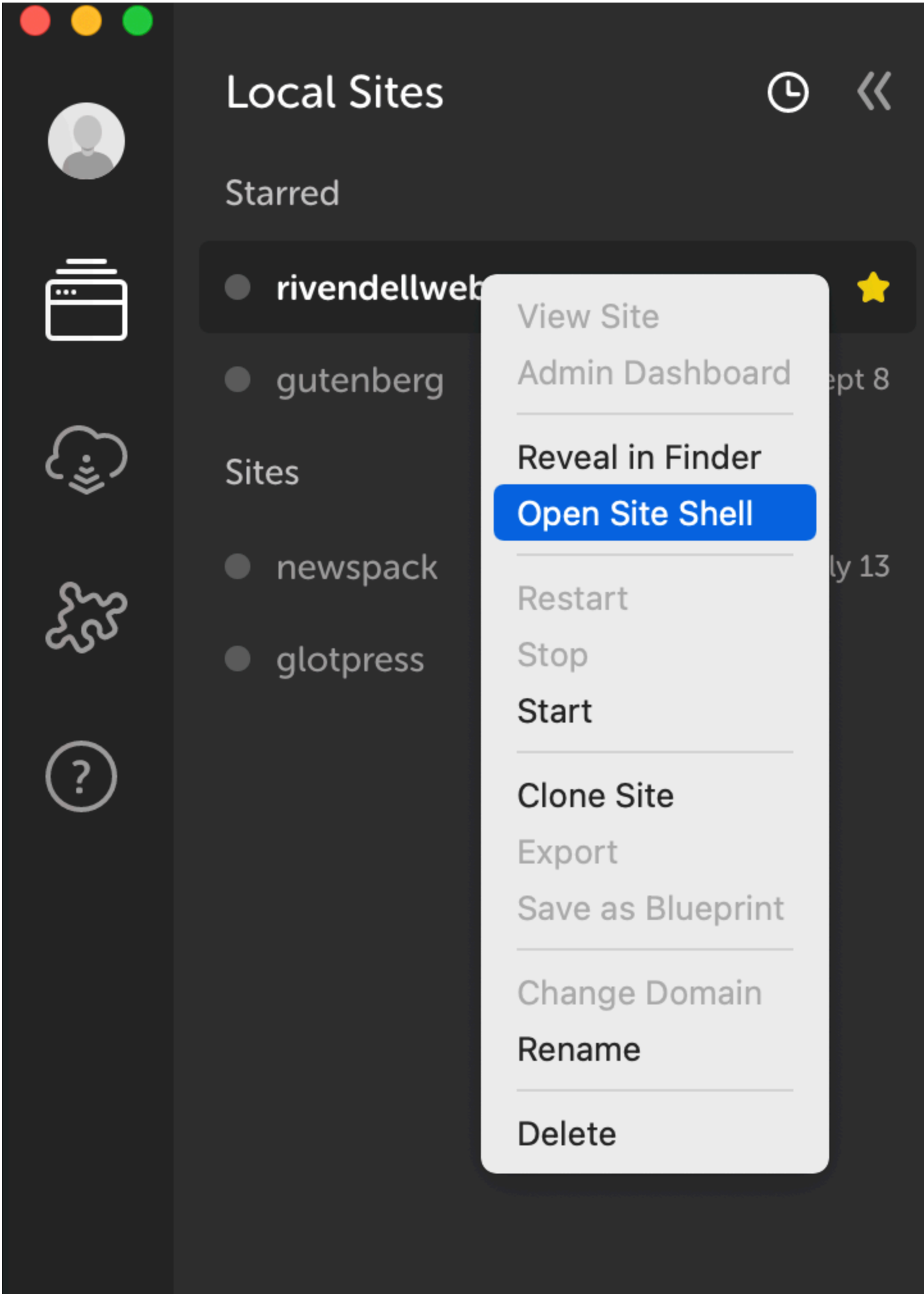
Local Sites
Starred
rivendellweb
gutenberg
Sites
newspack
glotpress
View Site
Admin Dashboard
Reveal in Finder
Open Site Shell
Restart
Stop
Start
Clone Site
Export
Save as Blueprint
Change Domain
Rename
Delete

Figure 1: selection of site shell for a local application

You will see a new terminal shell that looks like this:

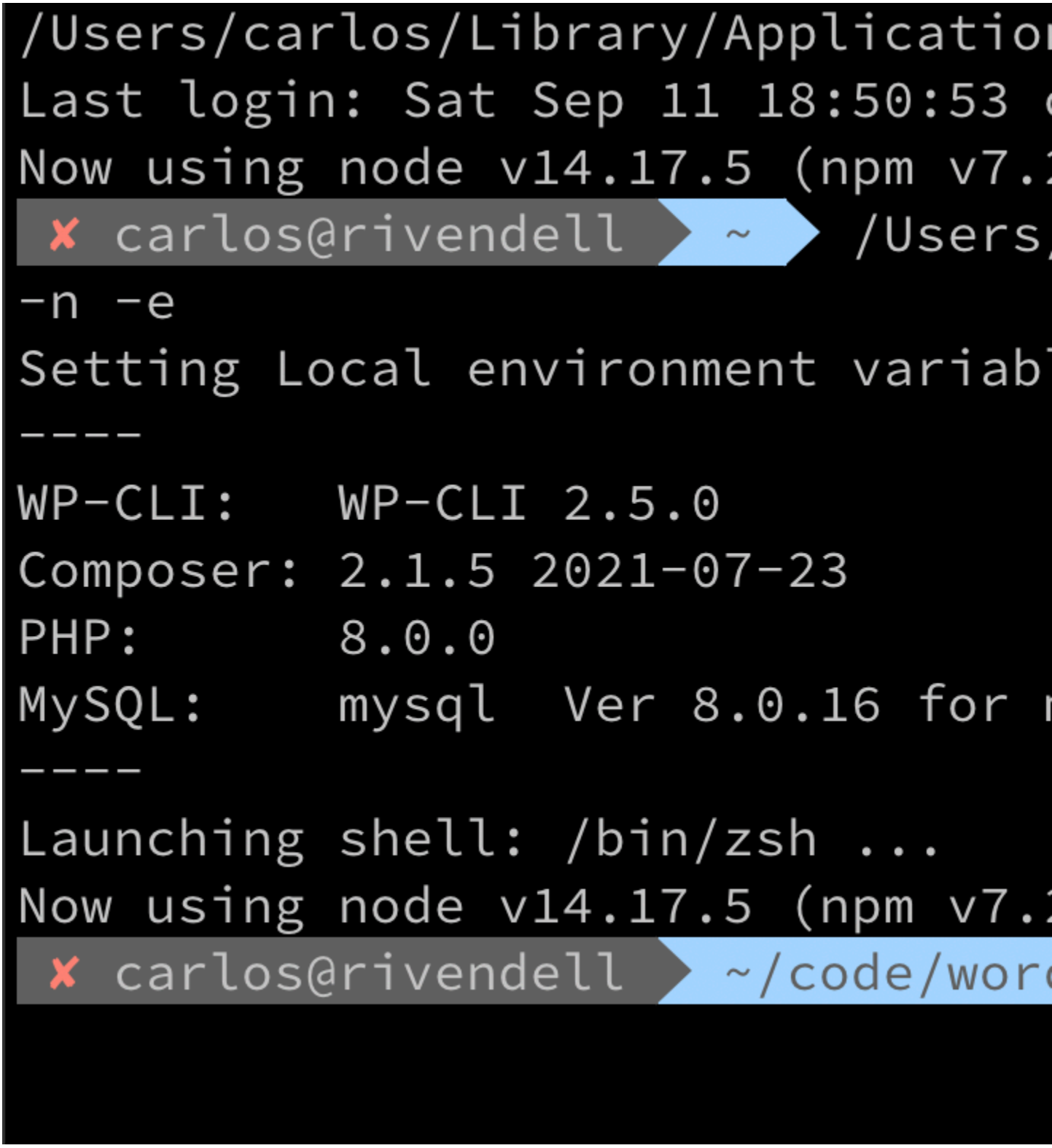

Figure 2: Shell opening for Local site

The shell comes preconfigured with the WordPress CLI and PHP Composer

WordPress CLI

The WordPress CLI gives you command line access to a WordPress site (either local or remote). For the purposes of this discussion we'll look at the installation provided by Local.

One of the most interestings things, to me, is that the CLI allows you to do a lot of things that would take a lot of time to do manually.

I will look at two examples of things that can be done with the CLI.

The i18n in the CLI handles workbloads for Javascript/React blocks, as documented in New! JavaScript i18n support in WordPress 5.0, and for PHP workloads.

- wp i18n make-json — Extract JavaScript strings from PO files and add them to individual JSON files
- wp i18n make-mo — Create MO files from PO files
- wp i18n make-pot — Create a POT file for a WordPress project

These commands work together with tools like POEdit to create language packs for your site.

Another thing that I like a lot is the set of scaffold commands. This set of commands, one of many, will create default elements using current WordPress standards and best practices.

The following commands are available:

- wp scaffold block — Generates PHP, JS and CSS code for registering a Gutenberg block for a plugin or theme
- wp scaffold child-theme — Generates child theme based on an existing

theme
- wp scaffold plugin — Generates starter code for a plugin
- wp scaffold plugin-tests — Generates files needed for running PHPUnit tests in a plugin
- wp scaffold post-type — Generates PHP code for registering a custom post type
- wp scaffold taxonomy — Generates PHP code for registering a custom taxonomy
- wp scaffold theme-tests — Generates files needed for running PHPUnit tests in a theme
- wp scaffold underscores — Generates starter code for a theme based on the Underscores (_s) starter theme
- wp scaffold _s — Generates starter code for a theme based on the Underscores (_s) starter theme

PHP Composer

The other command tool that Local makes available is PHP Composer (Composer for short), the PHP equivalent of NPM or Yarn in Node or Homebrew in macOS.

You can use Composer to install and manage dependencies for your PHP projects. This is less of an issue on the PHP side than it is using React and Gutenberg since there are far fewer dependencies to manage. Still it is nice to have, since sooner or later we'll want to run unit tests and other things that require external dependencies.

Conclusion

Composer is one of those nice-to-have tools until you need it and then you wonder how you could work without it.

We only scratched the surface of what the WordPress CLI can do... there is so much more to explore that covering all of it would make this post extremely long. I encourage you to explore the CLI and see what you can do with the commands in it.