MAINTAIN CONTROL OF YOUR CODE WITH GIT

WHITE PAPER

Jorge Perez, WP Engine Enterprise Support

August 2015

MAINTAIN CONTROL OF YOUR CODE WITH GIT

If you're a developer, you've probably heard a lot of your peers talking about Git. And all of this talk may have you wondering, "Hey, what the heck is Git anyway?"

You're not alone. Git is one of the many developer tools we offer here at WP Engine, and we're asked about it nearly every day. (Interesting bit of trivia: WP Engine was the first managed WordPress hosting provider to fully integrate git push into a hosted application.)

So, we decided we'd put together this white paper to give you the Git 411. We'll tell you what Git is, why it's important, key terms you should know, how WP Engine works with Git, and some common issues and troubleshooting. Once you've read this, you'll be ready to Git!

Let's Git started.

What Is Git?

Simply put, Git is a version control system. Sometimes, it's known as a Revision Control System. Git helps keep track of source code, or files, for programs and other projects, such as WordPress. One of the biggest advantages of Git is revisions—a developer can go back to an earlier version of the code if needed. It also lets you know who made changes to your code, what changes were made, and why those changes were made. Git's key attributes are speed, data integrity, and support for non-linear workflows. What it really comes down to is control.

A Git Command Glossary

Git, like any other technology, brings with it a set of terms and phrases you'll need to know to master it. Here's a lineup of some of the most common Git commands you should get familiar with.

Git repository (Git repo)

In order to use Git, you first have to have a Git repository, or Git repo. Simply put, a Git repository is a place where the history of your work is stored. It often lives in a .git subdirectory of your working copy—a copy of the most recent state of the files you're working on.

git init

This command initiates tracking of your project's source code and files within a repository. Once git init has run, Git can then do things like detect file changes, update remote repositories, and more from that time on.

git add

This command tells Git you'd like to add a file or files to the next snapshot that's created. It does not, however, make a new snapshot on its own. In Git terms, this means these files are staged for a commit.

git status

The git status command shows the status of the working tree. It displays any pending file changes, what files are ready to be committed, and more.

git commit

Using the git commit command lets you take a new snapshot of the repository, and developers can add comments for further details. Best practice dictates "commit often, perfect later."

git log

Using the git log command shows you the development history behind your most recent commit. Available options for this command will allow you to tailor the output, showing as much information as needed.

git remote

This command allows you to manage the set of repositories (or "remotes") whose branches you track.

git push

The git push command lets you update a remote repository with your changes. You can transfer files via Git using SSH, HTTP, or Git protocols.

Now that you know some of the most common Git commands, let's dig into how WP Engine uses Git.

How Does WP Engine Use Git?

At WP Engine, our managed WordPress hosting platform supports Git. Here are some ways you can use Git on the WP Engine platform:

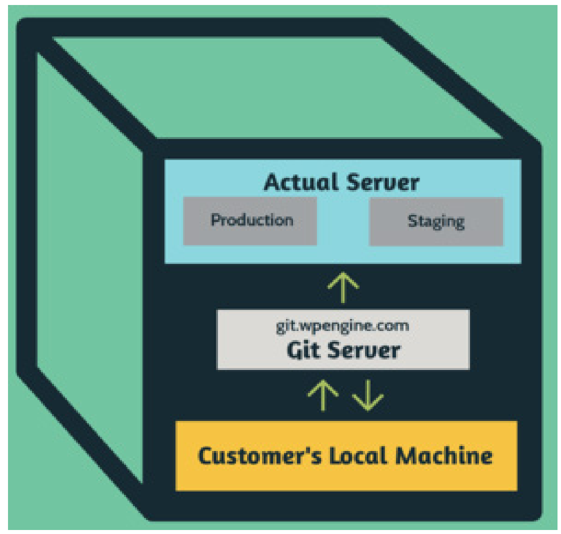
- Add your SSH key to the user portal. We support Git over SSH. Adding this key will work for authentication when you push your files to your WP Engine server.
- Download a backup and commit locally. This can be done using the Backup Points option in the user portal and then running git add and then git commit locally.
- Configure git remote to tell Git where to send files when pushing (git remote add), and which files Git should not track (i.e. .gitignore).
- Deploy code with Git. When code is ready to be updated, you can run the git push command to copy files from your machine to your WordPress install. You can do this either in live or staged environments.

It should be noted that at WP Engine, Git is a one-way transaction. We support git push functionality, so you can push your files to the WP Engine platform, however, we currently don't support git clone, and we're working on a way to fully support git pull (we kind of support it now). Stay tuned for more information as we add Git functionality to our growing set of developer tools.

When you're using Git to develop on the WP Engine platform, you never push code to your actual server—you push to the git.wpengine.com server, which in turn deploys the code to your actual server (this is why we don't currently support git clone). When the Git server deploys to the actual server, it runs an rsync command to sync any changed files.

A quick word about Git and SFTP: Git tracks files. If you upload a file via SFTP and then try to git push later, you're going to have a bad time, as code versioned through git push will overwrite code uploaded via SFTP. Stick to one or the other, we don't advise trying to do both.

Here's a look at the Git data flow at WP Engine:



When you use git push, the git server receives the content of the push. After the push has been processed, the Git server deploys the code to your actual server and environment to which you pushed it, whether that's production or staging.

Please note how data can be pushed to and pulled from the Git server. Data can never be pulled from the actual live server via Git. Again, this is why git clone does not provide updated site content on our platform.

A Typical Use Case For Git

Here's a rundown of one of the ways you could use Git on the WP Engine platform:

1. You make local changes to your WordPress theme, which is tracked by Git.
2. You run git add to tell Git, "Hey, when I make my next snapshot, I want to include all of my changed files."

```
git add .
```

3. You then run git commit to tell Git, “Hey, take a snapshot of how things are now” and you add a commit comment. Commits are stored in Git history so in the future you can see what was added to that snapshot, by whom, and for what reason.

```
git commit -m 'new styles for my theme'
```

4. You decide you are ready to test these changes on your site.
5. You use git push to put your file changes onto your staging site

```
git push staging master
```

6. If everything looks good, you use git push to put your file changes onto your production site.

```
git push production master
```

7. Everything is awesome and there are no problems.
8. We all celebrate!

Sometimes It Gits Tricky

There are some times when Git doesn’t perform exactly as you want it to. Let’s look at some common issues and how to troubleshoot them.

One issue with Git is that it’s challenging to replicate errors. This can happen for any number of reasons, one of which is that we would need a customer’s exact repo and SSH key to replicate an issue, and that’s just something we don’t do. We also rely on you for information regarding Git. The best way to deal with Git issues is to provide information—lots of information. Typically, when there’s a Git hiccup, we’re flying blind, so the more information you can share with us, the better. This will help us determine the root cause of the problem. We especially love it if you can provide us with your Git log using the command git log.

Note: when it comes to Git; logs and command output can be priceless. Information is king!

Common Issues

While Git is fairly easy to use, there are some issues that arise. Let‘s take a look at some of the most common ones:

- Git is asking for a password when you push: This could mean you either have not yet setup your SSH key, or that you added it and have not waited long enough for it to take effect (it takes roughly 10 minutes, but sometimes longer). It could also mean that you have more than one key set up locally. If that’s the case, you’ll need to set up an ssh_config file. If you encounter this on our platform, we’ll likely ask you to provide your local key’s fingerprint so we can compare it to the one in the user portal. If those don’t match, we’ve found your problem. If it does match, we’re back to the drawing board.
- “fatal: Could not read from remote repository:” This usually means one of two things: either you don’t have access for that install, or a bigger problem, which will require a ticket to get it fixed.
- “Everything up-to-date:” This means that the HEAD of the branch on the Git server is at the same commit ID as the local branch that is pushing to the server. Or, more plainly, this means that since both IDs are the same, Git responds by saying “Hey, we’re at the same place in our history, so there’s nothing you need to push,” and it stops receiving. It’s simple to fix: just open any file that you are pushing, add a comment or blank line, and save the file. Then run git add for that file, git commit, and then git push once more. Voila! Fixed!
- [rejected] non-fast forward updates rejected: Git is designed to look out for you. It performs a number of checks to ensure history integrity. If you try to push to the Git server and the repo has changes that you do not already have locally, you’ll receive this error. It’s Git’s way of telling you, “Hey, you are trying to make changes, but other changes were made since the last time you pushed. So, I’m going to have to reject this.” Just do a git pull to bring down those changes into your local copy, commit them, and then git push to your site.

Connecting GitHub Or BitBucket To WP Engine

Currently, there is no direct way to connect GitHub or BitBucket to WP Engine, however, there are two workarounds that can help:

- First, you can set up a third-party service that connects to your repo at GitHub or BitBucket (there are plenty of third-party services available). Once it’s connected to it, you can make or detect certain changes, like commits and hooks, and then SFTP those file changes to your account with WP Engine. That way, you can keep your normal workflow without having to change much on your end to integrate Git with the WP Engine platform.
- Second, you can create a workflow that designates one person to push changes to the WP Engine server, like so:
  - That person runs git pull to pull files down from your remote repo at GitHub or BitBucket. Once you git pull, all files pulled into that remote repo will live on your local machine.
  - You can then run git push to the Git server at WP Engine, and it will go through the deploy process.
  - As a shortcut, you can alias a command to do something like “alias wpgit=”git pull production master && git push wpengine master.”

Go On, Git!

That about sums up Git and how to use it on the WP Engine platform. You'll see that Git is a great service for working and collaborating with other developers, while allowing you to maintain a strong level of control.

Git is just one of the many innovative, developer-friendly tools WP Engine offers to make your life and your job easier. For more information, check us out at wpengine.com.

About WP Engine

WP Engine is a leading SaaS content management platform for websites and applications built on WordPress. The company powers thousands of websites and apps built on the WordPress platform delivering a fast, reliable and secure web experience. All levels of users including bloggers, marketers, SMBs and large corporations rely on WP Engine's platform to keep their websites up and running. The company's exceptional customer service team specializes in quickly solving technical problems, and creating a world-class customer experience ensuring that each user's WordPress site continues to perform at its full potential. Founded in 2010, WP Engine is headquartered in Austin, Texas and has offices in San Francisco, California, San Antonio, Texas and London, England.