

DEPT.

Using Github Actions for continuous deployment

You like to keep all your Drupal website code in Github (GUIs for PRs! Clear code diffs! Searchability! Visibility within your organization!), but hosting partners like Pantheon and Acquia have their own Git repositories that tie into their deployment processes. This session will walk you through taking an existing Github Drupal repository and automatically deploying any updates to that repository's main branch (eg, by merging in a pull request) to a Pantheon dev environment.

(As of August 2023) Github accounts include 2000 minutes of Github actions execution minutes for private repositories, and Pantheon's Silver plan is also free. So if you'd like to reproduce what I'm showing to explore the possibilities for your own websites, you can do so at no cost.

DEPT.

Introduction



Gena "Jenna" Wilson

Principal Software
Engineer (Drupal)
at DEPT®

github.com/genamw

Introduction: My name is Gena Wilson, I'm a Principal Software Engineer at Dept, a full-service digital agency with over 30 offices around the world. They do lots of work that is not Drupal, but I'm the resident Drupal enthusiast in the United States. I only work on Drupal projects and infrastructure, and I'm excited to be here at Twin Cities Drupal Camp with people who also work with Drupal.

I've been with Drupal since Drupal 6 in 2011, started with it on volunteer website builds and eventually it became my day job. I've been working with it long enough that when people talk through technical problems, my mind goes to how to accomplish those things in Drupal.

DEPT.

Agenda

- Initialize a new Drupal site locally (with DDEV) and put it on Github
- Modify our new Drupal site so it works with Pantheon
- Push our site to Pantheon
- Verify

Agenda (Here's what we'll do together)

- Initialize a new Drupal site locally (with DDEV) and put it on Github
- Modify our new Drupal site so it works with Pantheon
- Push our site to Pantheon
- Verify

DEPT.

Caveats and Disclaimers

- This is the simplest use case
- I am not a devops person
- I am not a git expert

Caveats and disclaimers

- This is the simplest use case
 - no multidev
 - only pushes to dev environment
 - it assumes you have *all* the permissions on both Github and Pantheon
- I am not a devops person (I am a Drupal developer doing my best)
- I am not a git expert

I am here today to share "a" way that works to accomplish continuous deployment (not "the way" or "the only way"). Take what's useful to you from it, and build on it and extend it in a way that works for you. :)

DEPT.

Prepare Pantheon

- Create a Pantheon sandbox site
- Generate and add SSH keys
- Generate (and save) a Terminus Machine Token

Create a Pantheon sandbox site

Let's get Pantheon set up.

<https://dashboard.pantheon.io/>

1) Log into Pantheon. Create a sandbox site. Choose Drupal 10.
(this takes a long time. do not do it in realtime)

It will have an update. apply it

Put it/leave it in Git mode (not SFTP).

Add SSH keys

- Add SSH key <https://docs.pantheon.io/ssh-keys>
- Generate one for Github and add it to your account, or if you have the users, create a Github user and add it to that user's account.
(I'm doing this for free, so my SSH key here is tied to my Pantheon user account. If you can, create a deployment user instead.)
- `ssh-keygen -t rsa -C "Github-actions@github.local"`
 - (Put it on the Desktop for later)
- `pbcopy < tcdrupal_rsa`

Generate a Terminus Machine Token

- Terminus Machine token <https://docs.pantheon.io/machine-tokens>
- Write it down / put it in a password manager because it only appears once!
- Recommend naming it for Github

DEPT.

Initialize our
new Drupal site
(or prepare the
old one)

- Do not commit the files that Composer pulls in (modules/contrib, core, etc)
- Have a healthy gitignore that enforces this
- Document how to build locally in your ReadMe.md
- Add Drush

Initialize our new Drupal site:

Goals:

- Do not commit the files that Composer pulls in (modules/contrib, core, etc)
- Have a healthy gitignore that enforces this
- Document how to build locally in your ReadMe.md
- Add a site-specific Drush

Alternatively:

Start with Pantheon's repo

DEPT.

Initialize our
new Drupal site
(or prepare the
old one)

<https://ddev.readthedocs.io/en/latest/users/quickstart/#drupal>

Initialize a new Drupal site with DDEV and load it locally

I am going to start with a totally new site using a ddev quickstart recipe for D10.

<https://ddev.readthedocs.io/en/latest/users/quickstart/#drupal>

Prerequisites

- * Git must be installed (On my mac I use Xcode Command Line Tools)
- * A local development environment tool like ddev, lando, docksal is helpful

If ddev complains about ports, edit `.ddev/config.yaml` in your new site so it uses unused ports for http/https.

DEPT.

Personalize our
new Drupal site

- Make a theme change
- Install admin_toolbar
- Move site configuration from the files directory

Personalize our new Drupal site:

I will not spend much time here because you will hopefully already have a site you're planning to update, but I'll make 3 changes:

1) edit Olivero so it doesn't move (because I find that annoying)

* Appearance > Settings > Olivero > check "Enable mobile menu at all widths"

2) install admin_toolbar

* `ddev composer require drupal/admin_toolbar`

* `ddev drush en admin_toolbar admin_toolbar_tools`

3) move the site configuration from the files directory to config outside the web root (web)

* Edit settings.php so this line is uncommented (search for `config_sync_directory`), and set it to this:

** `$settings['config_sync_directory'] = '../config';`

** Run `> ddev drush cex` to export this site config to that directory and verify that it's in the right directory

DEPT.

Put our site on
Github

- Create a new repository, then...
- `git init -b main`
- `git remote add origin
git@github.com:user@reposito
ry.git`
- `git add .`

Now I'll create a new Github repository for this local site.

- 1) Go to github.com and log in.
- 2) Go to your repositories page; I got there by clicking on my profile image and selecting "Your repositories", "New"
- 3) Give it a name, select public or private, uncheck adding any files so you have an empty new repository

Now let's get the code into Github.

- 1) In your local site directory, initialize a git repo
 - > `git init -b main`
- 2) add the remote
 - > `git remote add origin git@github.com:git_user/repository_name.git`
- 3) Add and commit your local files
 - > `git add .`

Too many files!

DEPT.

Personalize our
new Drupal site
(continued)

- Too many files!
- Include .gitignore

Personalize: Include .gitignore

Add a .gitignore that excludes files generated with composer from your repository. (So the list of files in your repo should just be:

- * .gitignore
- * composer.json and composer.lock
- * config
- * web/sites/default/settings.php

DEPT.

Put our site on
Github
(continued)

- `git reset .`
- `git add .`
- `git commit -m "First commit"`
- `git push -u origin main`

```
git reset .  
git add .  
git commit -m "First commit"  
git push -u origin main
```

So this gets us a site that is composer-managed, a small repository with none of the composer-generated files checked in.

DEPT.

Put our site on
Github
(continued)

Add a Readme.md file

Github last step: Add a readme

This readme should include all steps from cloning the repository to viewing a fully functional local copy of the site.

DEPT.

Backup/Export
everything!

```
drush archive:dump
```

OR, for database only

```
ddev export-db
```

Backup/Export everything!

So it can be imported into Pantheon.

Export your site database and files in case anything from this point on goes wrong.

There are many ways to do this. Here is one way:

- Backup database: `ddev export-db`
`--file=/Users/me/Desktop/pantheonsite.sql.gz`
- Backup files: [TBD]
- OR, create site archive: `drush archive:dump`
`--destination=drush-archive.tar.gz`
- https://www.drush.org/12.x/commands/archive_dump/

If you have a lot of files, you'll probably need to use Terminus instead.

- <https://docs.pantheon.io/terminus/commands/import-database>
- <https://docs.pantheon.io/terminus/commands/import-files>
- <https://docs.pantheon.io/terminus/commands/import-site>

DEPT.

Modify our site
so it's ready for
Pantheon

Options:

- Git merge the Pantheon repo onto our Github repo
- Clone the Pantheon repo and merge your Github repo into it
- **Manually apply changes**

Options:

- Git merge the Pantheon repo onto our Github repo
 - `git remote add pantheon [your pantheon git url]`
 - `git merge pantheon/master --allow-unrelated-histories`
- Clone the Pantheon repo and merge your Github repo into it
- **Manually apply changes**

DEPT.

Manually
modify our site
so it's ready for
Pantheon

- cweagans/composer-patches
- Pantheon
upstream-configuration
- Pantheon yaml files
- Pantheon's settings.php

Manually modify our site so it's ready for Pantheon (September 2023 process)

1. Enable patching via cweagans/composer-patches
 - `composer require cweagans/composer-patches:^1.7`
2. Preserve upstream changes (this preserves Pantheon one-click updates...if you want that)
 - Copy upstream-configuration folder over from pantheon
 - Add Pantheon upstream-configuration folder location to composer.json
 - ```
{
 "type": "path",
 "url": "upstream-configuration"
}
```
  - `composer update --lock`
  - `composer require pantheon-upstreams/upstream-configuration:dev-main`
  - `composer require pantheon-systems/drupal-integrations:^10`
3. Pantheon yaml files: as siblings to composer.json, add:
  - pantheon.yml
  - pantheon.upstream.yml
4. Pantheon's settings.php updates
  - `web/sites/default/settings.pantheon.php`
  - Edit `web/sites/default/settings.php` to include above^ file

**DEPT.**

Verify and  
commit

Verify locally that our site still  
works.

Make sure your local site still works after these changes, updates to settings.php can cause disasters.

If it does, go ahead and push to Github!



**DEPT.**

Push to  
Pantheon

Import our backed-up database  
and files

Push to Pantheon 1: Import our DB (for this test site, we only have a DB)

DEPT.

Push to  
Pantheon

- Configure the Github Actions workflow file and add it to our Github repository
- Add Github repository secrets
- Commit the Github actions workflow file to our main branch

Next steps:

Configure the Github Actions workflow file and add it to our Git repository

(Talk through `./github/workflows/deploy-to-pantheon.yml`)

Add Github repository secrets

- <https://docs.github.com/en/actions/security-guides/encrypted-secrets#creating-encrypted-secrets-for-a-repository>
- Pantheon SSH key: `PANTHEON_SSH_PRIVATE_KEY`
  - `cd ~/Desktop`
    - (it should be on the Desktop from 100 slides ago when we prepared Pantheon)
  - `pbcopy < tcdrupal_rsa`
- Terminus machine token: `TERMINUS_TOKEN`

Commit the Github Actions workflow file to your Git repository's main branch

It will run automatically. Hopefully, it succeeds.

**DEPT.**

## Test and Verify

- Commit a new code change to our Github repository's main branch
- Follow progress on Github's repository Actions

Test and Verify:

Commit a new code change to our Github repository's main branch

Follow progress on Github's repository Actions

We are committing directly, but this same process will happen after merging in a pull request.

**DEPT.**

## Troubleshooting

- Watch the Github action process logs
- Put your troubleshooting commands in the workflow file
- Comment out commands and then re-enable them one at a time, to identify the one causing problems

## Troubleshooting

- You may want to turn off Upstream updates if you are continuing to maintain this site
- Watch the Github action logs
- Put local troubleshooting commands in your workflow file: git status, git remote -v, etc.
- Comment out commands with errors, sometimes that green check is a mood booster even if it's not doing what you set out to have it do

**DEPT.**

## Opportunities beyond

- Implement multidev with your Github branches
- Add in testing
- Add in front end asset compilation
- BLT <https://docs.acquia.com/blt/>

Opportunities beyond this walkthrough:

- Implement multidev with your Github branches!  
<https://docs.pantheon.io/guides/git/collaborative-development#feature-branching>
- Add in testing (CI)
- Add in front end asset compilation like compiling SASS into CSS
- BLT <https://docs.acquia.com/blt/>

DEPT.

# Thank you!

22-03-2021

Knock knock?

Who's there?

Git

Git who?

Git outta here to the Unconference with my thanks for attending this session!

**DEPT.**

## Learnings

- Drush Deploy instead of drush cr; drush updb -y; drush cim -y; drush cr;
  - <https://www.drush.org/11.x/deploycommand/>
- Parameter order matters for ssh-keyscan
- Before forcing the git push, even file permissions will diverge the histories, e.g.
  - diff --git  
a/web/sites/default/settings.php  
b/web/sites/default/settings.php  
old mode 100644  
new mode 100755

**DEPT.**

## Resources

- <https://gist.github.com/daggerhart/72563179b1ef0774433e2b71858f1781>
- <https://medium.com/swlh/pantheon-and-github-actions-automated-deployments-via-github-actions-c245aa954797>
- <https://medium.com/bluecadet/pantheon-and-github-actions-1b9b1dbd5746>
- <https://docs.pantheon.io/guides/git/collaborative-development>
- [https://stackoverflow.com/questions/52313505/cannot-push-commit-to-a-newly-created-repository-on-github \("warning: no common commits"\)](https://stackoverflow.com/questions/52313505/cannot-push-commit-to-a-newly-created-repository-on-github-warning-no-common-commits)
- <https://lastcallmedia.com/blog/improving-our-production-drupal-deployments-pantheon>