Continuous Deployment from Github to WP Engine

Zachary Watkins
About me

• WordPress Developer for AgriLife Communications
• JavaScript, PHP, front end, and dev ops
• Written code for thousands of sites
• B.F.A. in studio art
What is Continuous Deployment?

• A workflow that minimizes the amount of time needed to provide users with the latest version of a code base
• Frequent commits to version control
• Automated tasks, deployment, and feedback
Benefits

• Earlier return on investment
• Earlier feedback from users
• Faster response to users
• Automated deployment
• Increased productivity

1 https://www.agilealliance.org/glossary/continuous-deployment/
Deployment Cycle
Services and software used

• Git client – Sourcetree
• Task runner – Grunt
• Package manager – npm and Bower
• Repository hosting – Github and Bitbucket
Services and software used, cont.

• Browser and device tests – Browserstack
• Continuous deployment service – Codeship
• Web hosting – WP Engine
• Notifications – Slack
Staging Deployment Cycle
Write code locally
Local Development

• WordPress virtual server using VVV
• Browserstack
• Sass source maps
Commit, then push to staging branch
Commit and push

• If first commit after a release, also update version number in `package.json`
• Include `[skip ci]` in commit message to avoid triggering Codeship
Slack notifies GitHub
Slack channel notification
Codeship triggers and executes

GitHub

WPengine

npm

GRUNT
Codeship execution

- Clone staging branch (automatic)
- Run custom bash commands
  - Address shallow clone limitation
  - Set up repository info
  - Rearrange files
  - Import modules
  - Lint/Compile CSS/JS files, with source maps
  - Commit and force push
Custom bash commands

```bash
# Get around shallow update restriction
if [ ! -f ${HOME}/clone/.git/shallow ]; then git fetch --unshallow; fi

# Add User Data
git config --global user.name "codeship-agriflex3"
git config --global user.email "zachary.watkins@ag.tamu.edu"

# Add servers
git remote add servers $AGRILIFE_STAGING
git remote set-url --add --push servers $AGRILIFE_STAGING
git remote set-url --add --push servers $COALS_STAGING
git remote set-url --add --push servers $COUNTIES_STAGING
git remote set-url --add --push servers $AGRILIFEEXTENSION_STAGING
git remote set-url --add --push servers $TXMG_STAGING
git remote set-url --add --push servers $TXMN_STAGING

# Install needed modules
npm install -g grunt-cli
npm install -g grunt@0.4.0
npm install -g bower
npm install -g ruby
gem install compass

# Move repo files to a named folder
mkdir $FOLDERNAME
shopt -s extglob
mv !($FOLDERNAME) $FOLDERNAME
```
Custom bash commands, cont.

```bash
# Move repo files whose name begins with a period
mv .sass-lint.yml $FOLDERNAME/.sass-lint.yml
# Exclude development-only files from commit
rm .gitignore
mv .codeshipignore $FOLDERNAME/.gitignore
# Move named folder into a structure identical to the
# root directory of a WordPress server
mkdir -p $DIRECTORY
mv $FOLDERNAME $DIRECTORY
cd $DIRECTORY/$FOLDERNAME/
# Build
composer install
npm install
bower install html5shiv@3.7.0
bower install respond@1.4.1
grunt develop
# Deploy
git add --all :/
git commit -m "DEPLOYMENT"
git push servers HEAD:master --force
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td># RECOMMENDED BY WPENGINE</td>
</tr>
<tr>
<td>2</td>
<td>*~</td>
</tr>
<tr>
<td>3</td>
<td>.DS_Store</td>
</tr>
<tr>
<td>4</td>
<td>.svn</td>
</tr>
<tr>
<td>5</td>
<td>.cvs</td>
</tr>
<tr>
<td>6</td>
<td>*.bak</td>
</tr>
<tr>
<td>7</td>
<td>*.swp</td>
</tr>
<tr>
<td>8</td>
<td>Thumbs.db</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td># large/disallowed file types</td>
</tr>
<tr>
<td>11</td>
<td># a CDN should be used for these</td>
</tr>
<tr>
<td>12</td>
<td>*.hqx</td>
</tr>
<tr>
<td>13</td>
<td>*.bin</td>
</tr>
<tr>
<td>14</td>
<td>*.exe</td>
</tr>
<tr>
<td>15</td>
<td>*.dll</td>
</tr>
<tr>
<td>16</td>
<td>*.deb</td>
</tr>
<tr>
<td>17</td>
<td>*.dmg</td>
</tr>
<tr>
<td>18</td>
<td>*.iso</td>
</tr>
<tr>
<td>19</td>
<td>*.img</td>
</tr>
<tr>
<td>20</td>
<td>*.msi</td>
</tr>
<tr>
<td>21</td>
<td>*.msp</td>
</tr>
<tr>
<td>22</td>
<td>*.msm</td>
</tr>
<tr>
<td>23</td>
<td>*.mid</td>
</tr>
<tr>
<td>24</td>
<td>*.midi</td>
</tr>
<tr>
<td>25</td>
<td>*.kar</td>
</tr>
<tr>
<td>26</td>
<td>*.mp3</td>
</tr>
<tr>
<td>27</td>
<td>*.ogg</td>
</tr>
<tr>
<td>28</td>
<td>*.m4a</td>
</tr>
<tr>
<td>29</td>
<td>*.ra</td>
</tr>
<tr>
<td>30</td>
<td>*.3gpp</td>
</tr>
<tr>
<td>31</td>
<td>*.3gp</td>
</tr>
<tr>
<td>32</td>
<td>*.mp4</td>
</tr>
<tr>
<td>33</td>
<td>*.mpeg</td>
</tr>
<tr>
<td>34</td>
<td>*.mpg</td>
</tr>
<tr>
<td>35</td>
<td>*.mov</td>
</tr>
<tr>
<td>36</td>
<td>*.webm</td>
</tr>
<tr>
<td>37</td>
<td>*.flv</td>
</tr>
<tr>
<td>38</td>
<td>*.m4v</td>
</tr>
<tr>
<td>39</td>
<td>*.mng</td>
</tr>
<tr>
<td>40</td>
<td>*.asx</td>
</tr>
<tr>
<td>41</td>
<td>*.asf</td>
</tr>
<tr>
<td>42</td>
<td>*.wmv</td>
</tr>
<tr>
<td>43</td>
<td>*.avi</td>
</tr>
<tr>
<td>44</td>
<td></td>
</tr>
</tbody>
</table>
.codeshipignore, cont.

<table>
<thead>
<tr>
<th>Line</th>
<th>File</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td># Don't deploy</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>package.json</td>
<td>65 bower_components/**/foundation/scss/</td>
</tr>
<tr>
<td>47</td>
<td>bower.json</td>
<td>66 bower_components/**/jquery/src/</td>
</tr>
<tr>
<td>48</td>
<td>.bower.json</td>
<td>67 bower_components/<strong>/modernizr/</strong>/</td>
</tr>
<tr>
<td>49</td>
<td>.bowerrc</td>
<td>68 bower_components/**/modernizr/grunt.js</td>
</tr>
<tr>
<td>50</td>
<td>config.rb</td>
<td>69 bower_components/**/picturefill/index.html</td>
</tr>
<tr>
<td>51</td>
<td>node_modules</td>
<td>70 bower_components/**/picturefill/logos/<em>.</em>.png</td>
</tr>
<tr>
<td>52</td>
<td>.sass-cache</td>
<td>71 bower_components/**/superfish/examples/</td>
</tr>
<tr>
<td>53</td>
<td>*.md</td>
<td>72 bower_components/html5shiv/</td>
</tr>
<tr>
<td>54</td>
<td>*.txt</td>
<td>73 !bower_components/html5shiv/dist/html5shiv.js</td>
</tr>
<tr>
<td>55</td>
<td>*.ai</td>
<td>74 css/src</td>
</tr>
<tr>
<td>56</td>
<td>*.scss</td>
<td>75 js/src</td>
</tr>
<tr>
<td>57</td>
<td>*.coffee</td>
<td>76</td>
</tr>
<tr>
<td>58</td>
<td>.gitignore</td>
<td>77 # Prevent Composer files from being loaded if they are only used for</td>
</tr>
<tr>
<td>59</td>
<td>LICENSE</td>
<td>78 # PHP5.6+</td>
</tr>
<tr>
<td>60</td>
<td>LICENSE-MIT</td>
<td>79 # vendor/composer/autoload_static.php</td>
</tr>
<tr>
<td>61</td>
<td>gruntfile.js</td>
<td>80</td>
</tr>
<tr>
<td>62</td>
<td>Gruntfile.js</td>
<td>81 # Ignore release file</td>
</tr>
<tr>
<td>63</td>
<td>werker.yml</td>
<td>82 AgriFlex3.zip</td>
</tr>
<tr>
<td>64</td>
<td>.editorconfig</td>
<td>83</td>
</tr>
</tbody>
</table>
grunt “develop” task: sasslint

- Enforce style guide for our sass files
grunt “develop” task: compass:dev

- Compile sass to css, expanded and with sourcemaps
- Include the Foundation import path

```javascript
  tasks: [ 'develop' ]
  compass:
    pkg:
      options:
        config: 'config.rb'
        force: true
    dev:
      options:
        config: 'config.rb'
        force: true
        outputStyle: 'expanded'
        sourcemap: true
        noLineComments: true
  ``
grunt “develop” task: jsvalidate

• Check js files for syntax errors

```javascript
grunt.initConfig({
  jsv: {
    files: {
      src: [
        'js/*.js',
        'bower_components/foundation/js/vendor/fastclick.js',
        'bower_components/foundation/js/foundation/foundation?(.topbar).js',
        'bower_components/modernizr/modernizr.js',
        'bower_components/jquery/{dist,sizzle}/**/.*.js',
        'bower_components/jquery-placeholder/*.js',
        'bower_components/jquery.cookie/jquery.cookie.js',
        'bower_components/respond/{cross-domain,dest}/**/.*.js',
        'bower_components/html5shiv/dist/html5shiv.js'
      ]
    },
    options: {
      globals: {
        jQuery: true,
        console: true,
        module: true,
        document: true
      },
      noEmpty: false,
      noLineComments: true
    }
  }
});
```
If passing: deploy to staging, notify
Slack channel notification

[Image of a Slack channel with a notification about new commits and builds]

- **GitHub BOT** 10:00 AM: [AgriFlex3:staging] 3 new commits by Zachary Watkins:
  - dd57890 Improve release message scripts for code readability - Zachary Watkins
  - 5d77561 Applying sass-lint ignore statements for the no-important rule - Zachary Watkins
  - 15b4a89 Update State Link Policy URL - Zachary Watkins

- **Codeship BOT** 10:09 AM: Build succeeded
  - Update State Link Policy URL
  - Branch: staging

- **GitHub BOT** 10:19 AM: Message #agriflex3
Perform manual tests
Manual Tests

• Double-check that changes are working on staging sites
• Sass source maps expedite troubleshooting and revisions
If failing: troubleshoot and restart
Reasons it might fail

• PHP files incompatible with server
• Grunt tasks
Reasons it might fail, cont.

- Module(s) causing errors

```javascript
Running "sasslint:target" (sasslint) task
Warning: /home/rof/src/github.com/AgriLife/AgriFlex3/wp-content/themes/AgriFlex3/node_modules/grunt-sass-lint
/node_modules/sass-lint/node_modules/eslint/lib/formatters/stylish.js:7
const chalk = require("chalk"),
^^^
Use of const in strict mode. Use --force to continue.

Aborted due to warnings.
```
Production Deployment Cycle
Merge staging branch, then push
Merge and push

- If using [skip ci] in commit(s) to merge, and don’t want to deploy, then check box to include previous commit messages
Slack notifies GitHub
Slack channel notification
Codeship triggers and executes
Codeship execution

• Clone production branch (automatic)
• Run custom bash commands
  • Address shallow clone limitation
  • Set up repository info
  • Rearrange files
  • Import modules
• Lint/Compile CSS/JS files, compressed
• Release
• Commit and force push
Custom bash commands

```bash
# Get around shallow update restriction
if [ -f ${HOME}/clone/.git/shallow ]; then git fetch --unshallow; fi

# Add User Data
git config --global user.name "codeship-agriflex3"
git config --global user.email "zachary.watkins@ag.tamu.edu"

# Add servers
git remote add servers $AGRILIFE_PRODUCTION
git remote set-url --add --push servers $AGRILIFE_PRODUCTION
git remote set-url --add --push servers $COALS_PRODUCTION
git remote set-url --add --push servers $COUNTIES_PRODUCTION
git remote set-url --add --push servers $AGRILIFE_EXTENSION_PRODUCTION
git remote set-url --add --push servers $TXMG_PRODUCTION
git remote set-url --add --push servers $TXMN_PRODUCTION

# Install needed modules
npm install -g grunt-cli
npm install -g grunt@0.4.0
npm install -g bower
npm install -g ruby
gem install compass

# Move repo files to a named folder
mkdir $FOLDERNAMESPACE
shopt -s extglob
mv !($FOLDERNAMESPACE) $FOLDERNAMESPACE
```
Custom bash commands, cont.

```bash
# Move repo files whose name begins with a period
mv .sass-lint.yml $FOLDERNAME/.sass-lint.yml

# Exclude development-only files from commit
rm .gitignore
mv .codeshipignore $FOLDERNAME/.gitignore

# Move named folder into a structure identical to the root directory of a WordPress server
mkdir -p $DIRECTORY
mv $FOLDERNAME $DIRECTORY
cd $DIRECTORY/$FOLDERNAME/

# Build
composer install
npm install
bower install html5shiv#3.7.0
bower install respond#1.4.1
grunt package

# Release
grunt release

# Deploy
git add --all :/
git commit -m "DEPLOYMENT"
git push servers HEAD:master --force
```
<table>
<thead>
<tr>
<th></th>
<th>RECOMMENDED BY WPENGINE</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td># large/disallowed file types</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>*~</td>
<td>24</td>
</tr>
<tr>
<td>3</td>
<td>.DS_Store</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>.svn</td>
<td>26</td>
</tr>
<tr>
<td>5</td>
<td>.cvs</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>*.bak</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>*.swp</td>
<td>29</td>
</tr>
<tr>
<td>8</td>
<td>Thumbs.db</td>
<td>30</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>31</td>
</tr>
<tr>
<td>10</td>
<td># a CDN should be used for these</td>
<td>32</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>33</td>
</tr>
<tr>
<td>12</td>
<td>*.hqx</td>
<td>34</td>
</tr>
<tr>
<td>13</td>
<td>*.bin</td>
<td>35</td>
</tr>
<tr>
<td>14</td>
<td>*.exe</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>*.dll</td>
<td>37</td>
</tr>
<tr>
<td>16</td>
<td>*.deb</td>
<td>38</td>
</tr>
<tr>
<td>17</td>
<td>*.dmg</td>
<td>39</td>
</tr>
<tr>
<td>18</td>
<td>*.iso</td>
<td>40</td>
</tr>
<tr>
<td>19</td>
<td>*.img</td>
<td>41</td>
</tr>
<tr>
<td>20</td>
<td>*.msi</td>
<td>42</td>
</tr>
<tr>
<td>21</td>
<td>*.msp</td>
<td>43</td>
</tr>
<tr>
<td>22</td>
<td>*.msm</td>
<td>44</td>
</tr>
</tbody>
</table>
.codeshipignore, cont.

```bash
# Don't deploy
package.json  bower_components/**/foundation/scss/
bower.json   bower_components/**/jquery/src/
.bower.json  bower_components/**/modernizr/**/
.bowerrc    bower_components/**/modernizr/grunt.js
config.rb   bower_components/**/picturefill/index.html
node_modules bower_components/**/picturefill/logos/**.png
.sass-cache bower_components/**/superfish/examples/
.postcss-cache bower_components/html5shiv/
.txt         !bower_components/html5shiv/dist/html5shiv.js
.md          css/src
.ai          js/src
.sass        
.coffee      
.gitignore  # Prevent Composer files from being loaded if
LICENSE     # they are only used for PHP5.6+
LICENSE-MIT # vendor/composer/autoload_static.php
Gruntfile.js
Gruntfile.js  # Ignore release file
werker.yml   AgriFlex3.zip
.editorconfig
```
grunt “package” task: compass:pkg

- Compile sass to css, compressed
- Include the Foundation import path

```javascript
tasks: ['develop']
compass:
  pkg:
    options:
      config: 'config.rb'
      force: true
  dev:
    options:
      config: 'config.rb'
      force: true
      outputStyle: 'expanded'
      sourcemap: true
      noLineComments: true
```
grunt “package” task: jsvalidate

• Check js files for syntax errors
If passing: upload release
Example release

Release

ZachWatkins released this 6 days ago · 6 commits to master since this release

Zachary Watkins (3):

- Improve release message scripts for code readability
- Applying sass-lint ignore statements for the no-important rule
- Update State Link Policy URL

Downloads

- AgriFlex3.zip (1.31 MB)
- Source code (zip)
- Source code (tar.gz)
Gruntfile.coffee: release task

@registerTask 'release', ['compress', 'setreleasemsg', 'gh_release']

• Create .zip of all non-development files
• Create release message
• Upload release to Github
grunt-contrib-compress module

• List of file paths, some using “glob” patterns

```javascript
target: [css/src/*.js, css/*/*(css)];
compress:
  main:
    options: 'AgriFlex3.zip'
    files: [
      {src: ['AgriFlex/*.php']},
      {src: ['css/*/*(css)']},
      {src: ['img/*/*]},
      {src: ['js/*/*']},
      {src: ['bower_components/fastclick/lib/fastclick.js']},
      {src: ['bower_components/foundation/*/*(css,js)']},
      {src: ['bower_components/modernizr/modernizr.js']},
      {src: ['bower_components/jQuery/dist, sizzle/*/*(js)']},
      {src: ['bower_components/jQuery-Placeholder/*(js)']},
      {src: ['bower_components/jQuery.Cookie/jquery.cookie.js']},
      {src: ['bower_components/respond/cross-domain,dest/*(js)']},
      {src: ['bower_components/html5shiv/dist/html5shiv.js']},
      {src: ['vendor/*/*, '!vendor/composer/autoload_static.php']},
      {src: ['functions.php']},
      {src: ['README.md']},
      {src: ['rtl.css']},
      {src: ['screenshot.png']},
      {src: ['search.php']},
      {src: ['style.css']},
    ]
```
Custom `setreleasemsg` task

- Store the repo’s last tag for use by the `shortlog` task
- Execute the `shortlog` task

```javascript
@registerTask 'Release', ['compress', 'setreleasemsg', 'gh_release']
@registerTask 'setreleasemsg', 'Set release message as range of commits', ->
  done = async()
  grunt.util.spawn {
    cmd: 'git'
    args: [ 'tag' ]
  }, (err, result, code) ->
    if result.stdout isnt ''
      matches = result.stdout.match /(\n)+$/
      grunt.config.set 'lasttag', matches[1]
      grunt.task.run 'shortlog'
  done(err)
  return
return
@registerTask 'shortlog', 'Set GitHub release body with commit messages since last release', ->
```
Custom shortlog task

• Get commit messages since last release
• Remove [skip ci] and make into bulleted list
• Assign to release message
grunt-gh-release module

- Github personal access token for release key
- Pull info from package.json
Result

**Release**

ZachWatkins released this 6 days ago • 6 commits to master since this release

Zachary Watkins (3):

- Improve release message scripts for code readability
- Applying sass-lint ignore statements for the no-important rule
- Update State Link Policy URL

**Downloads**

- AgriFlex3.zip (1.31 MB)
- Source code (zip)
- Source code (tar.gz)
If still passing: deploy, notify
Slack channel notification
Perform manual tests
If failing: troubleshoot and restart
Reasons it might fail

- PHP files incompatible with server
- Grunt tasks
Reasons it might fail, cont.

- Module(s) causing errors

```
Running "sasslint:target" (sasslint) task
Warning: /home/rof/src/github.com/AgriLife/AgriFlex3/wp-content/themes/AgriFlex3/node_modules/grunt-sass-lint
/node_modules/sass-lint/node_modules/eslint/lib/formatters/stylish.js:7
const chalk = require("chalk"),
^^^^
Use of const in strict mode. Use --force to continue.

Aborted due to warnings.
```
“Why should I do this again?”
Benefits

- Earlier return on investment
- Earlier feedback from users
- Faster response to users
- Automated deployment
- Increased productivity

1 https://www.agilealliance.org/glossary/continuous-deployment/
Questions?
Credits

• Continuous Deployment, agilealliance.org
• Alex Zuniga (WP Engine) for pointing me to an improved shallow push solution
Resources

- AgriFlex3 Github repository
- grunt-contrib-compress
- grunt-gh-release
- node-glob file path rules
- Continuously Deploy your Wordpress Theme to WP Engine with Codeship, by Roman Kuba