Campbell's Private Label CI/CD Process

Requirements

- Have a Bitbucket account
- Some knowledge of command line *(BASH or Windows for your OS)*

1. Set up your environment

Set up SourceTree (Resource)

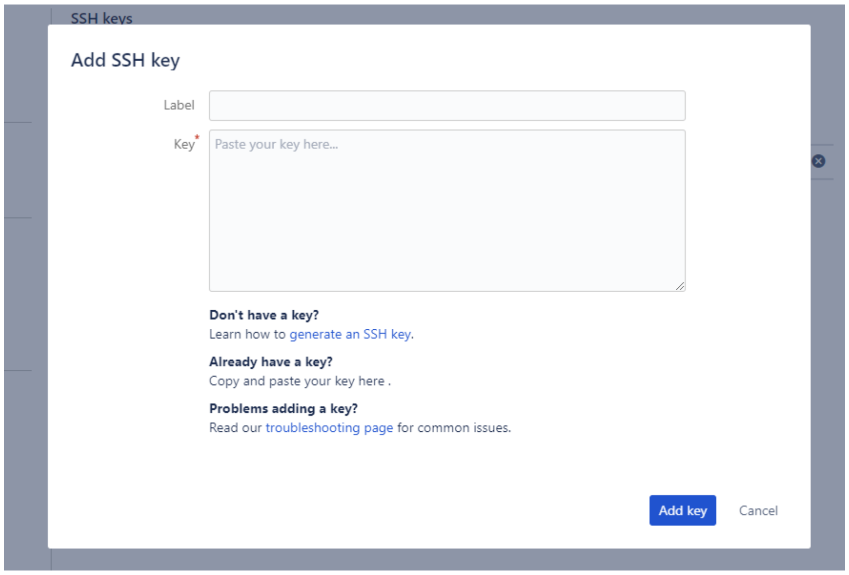
- Install SourceTree - https://confluence.atlassian.com/get-started-with-sourcetree/install-sourcetree-847359094.html

Create an SSH key for access (Resources)

- https://confluence.atlassian.com/bitbucketserver/creating-ssh-keys-776639788.html
- https://support.atlassian.com/bitbucket-cloud/docs/set-up-an-ssh-key/

Add SSH to your Bitbucket account

- Copy the content of the generated RSA file to your Bitbucket
    - Navigate to your account settings on Bitbucket > *Account > Settings > SSH Keys* – **Click on Add Key CTA** (Link - https://bitbucket.org/account/settings/)



Don't have a key?
Learn how to generate an SSH key.

Already have a key?
Copy and paste your key here .

Problems adding a key?
Read our troubleshooting page for common issues.

Configure SourceTree to use your SSH key

- This YouTube video demonstrates how it's done https://www.youtube.com/watch?v=640YX8TMdp4

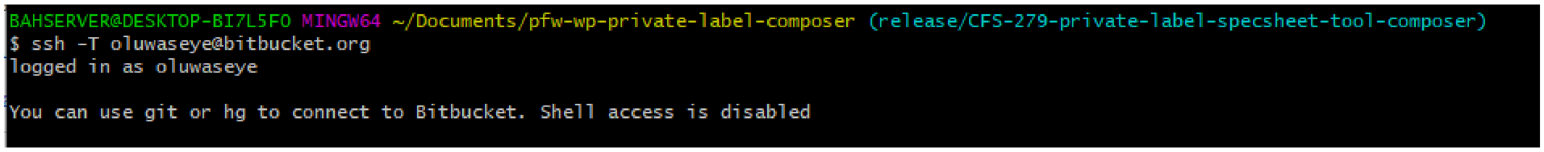
Install PHP & Composer

- You must have PHP on your machine. Composer is a dependency manager for PHP
- **Resource (Install PHP 7.4.4 on Windows 10)** https://www.youtube.com/watch?v=xAU3Tb7mm80
- **Resource (How to Install Composer on Windows 7/8/10)** https://www.youtube.com/watch?v=BGyuKpfMB9E

Test your SSH access and composer installation

- To test your SSH access, open the terminal (MAC/UNIX) or command line (windows)
- Since Campbell's Bitbucket is of type Git, use the following command
- `ssh -T username@bitbucket.org`

Screenshot



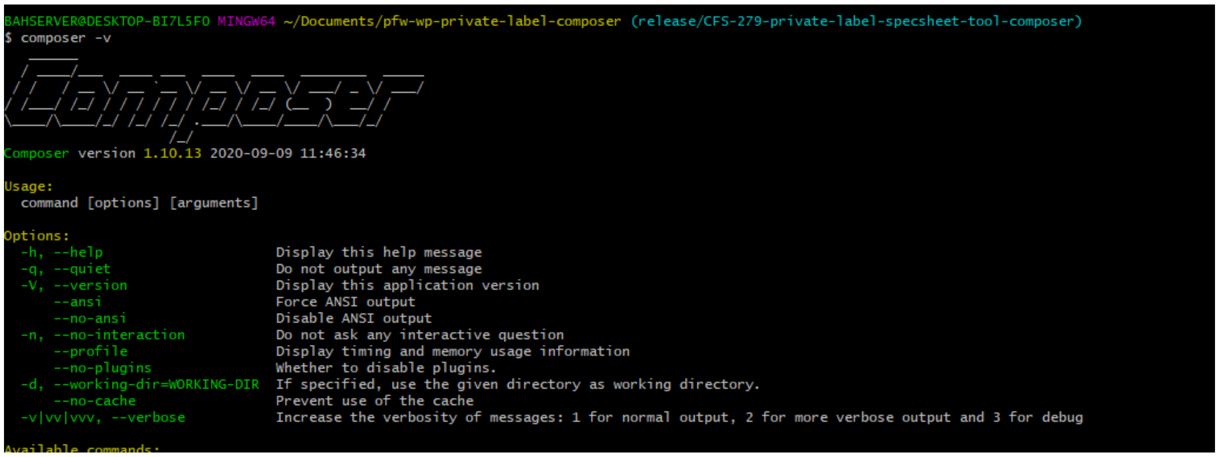
```
BAHSERVER@DESKTOP-BI7L5F0 MINGW64 ~/Documents/pfw-wp-private-label-composer (release/CFS-279-private-label-specsheet-tool-composer)
$ ssh -T oluwaseye@bitbucket.org
logged in as oluwaseye

You can use git or hg to connect to Bitbucket. Shell access is disabled
```

To test *composer*, simply check the version of *composer* by typing

`composer -v`

Screenshot



```
BAHSERVER@DESKTOP-BI7L5F0 MINGW64 ~/Documents/pfw-wp-private-label-composer (release/CFS-279-private-label-specsheet-tool-composer)
$ composer -v
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 1.10.13 2020-09-09 11:46:34

Usage:
  command [options] [arguments]

Options:
  -h, --help                     Display this help message
  -q, --quiet                    Do not output any message
  -V, --version                  Display this application version
      --ansi                     Force ANSI output
      --no-ansi                  Disable ANSI output
  -n, --no-interaction           Do not ask any interactive question
      --profile                  Display timing and memory usage information
      --no-plugins               Whether to disable plugins.
  -d, --working-dir=WORKING-DIR  If specified, use the given directory as working directory.
      --no-cache                 Prevent use of the cache
  -v|vv|vvv, --verbose           Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
```

2. Update the Campbell's Private Label Composer repo

Two repositories have to be updated each time you push the ***project branch***

Note: *Project branch e.g* ***(release/CFS-279-new-private-label-spec-sheet-too)*** *. A new branch is created for either a ticket or project. The process is to push to the project branch and merge to either* ***(develop), (stage), or (master)*** *branches.*

Develop branch is for the dev website: https://dev.privatelabel.campbellsfoodservice.com/

Stage branch is for the staging website: https://stage.distributorlabel.campbellsfoodservice.com/

The Master branch is for the production website: https://distributorlabel.campbellsfoodservice.com/

a. Clone these repositories on your machine

1. pfw-wp-private-label-cfs-theme: this repo is for the theme, your work on the theme goes here
2. pfw-wp-private-label-composer: the composer.lock file goes here

A little brief on how composer is being used on Campbell's Private label Website

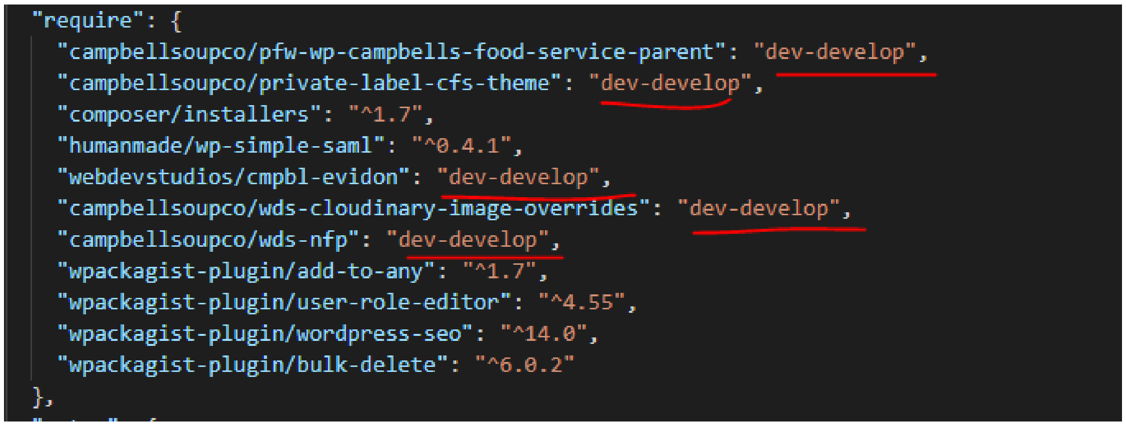
- Composer manages the state of most of the plugins (some created by WDS (Web DEV Studio), and others are regular WordPress plugins pulled from **https://wpackagist.org/**).

- ***Most importantly, **composer** manages the state of the theme in the theme repo **pfw-wp-private-label-cfs-theme** via the composer repo **pfw-wp-private-label-composer**.

  This means, every time you push to the project branch and merge into the develop branch, stage branch, or master branch of the theme repo, you have to update the composer repo right after that.

  The composer repo will update a lock file **(composer.lock)**. This lock file has to be pushed to the composer project branch of the composer repo and then merged into the develop branch, stage branch, or master branch of the composer repo.

  Example: After pushing an update to the project repo and merging to develop, you have to update your composer.json file to point to the develop branch (plugins and theme*).
  Screenshot



```
"require": {
    "campbellsoupco/pfw-wp-campbells-food-service-parent": "dev-develop",
    "campbellsoupco/private-label-cfs-theme": "dev-develop",
    "composer/installers": "^1.7",
    "humanmade/wp-simple-saml": "^0.4.1",
    "webdevstudios/cmpbl-evidon": "dev-develop",
    "campbellsoupco/wds-cloudinary-image-overrides": "dev-develop",
    "campbellsoupco/wds-nfp": "dev-develop",
    "wpackagist-plugin/add-to-any": "^1.7",
    "wpackagist-plugin/user-role-editor": "^4.55",
    "wpackagist-plugin/wordpress-seo": "^14.0",
    "wpackagist-plugin/bulk-delete": "^6.0.2"
},
```

Use this command to update the composer.lock file

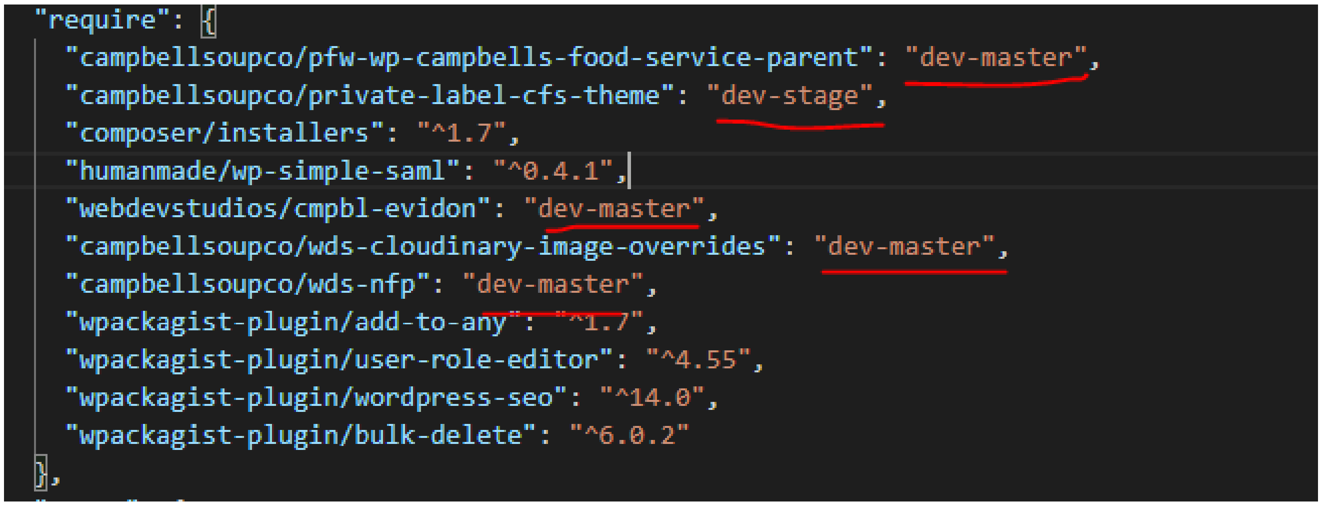
Next, you run the command with the composer repo.

```
composer update --no-dev
```

This generates a new composer.lock file for the first and updates an existing composer.lock file.

You only push the composer.lock file to the composer repo (project branch) > next you merge to develop.

When you are ready to move to the stage branch, update the composer.json's require object to point to dev-stage for the private label theme and dev-master for the plugins



```
"require": {
    "campbellsoupco/pfw-wp-campbells-food-service-parent": "dev-master",
    "campbellsoupco/private-label-cfs-theme": "dev-stage",
    "composer/installers": "^1.7",
    "humanmade/wp-simple-saml": "^0.4.1",
    "webdevstudios/cmpbl-evidon": "dev-master",
    "campbellsoupco/wds-cloudinary-image-overrides": "dev-master",
    "campbellsoupco/wds-nfp": "dev-master",
    "wpackagist-plugin/add-to-any": "^1.7",
    "wpackagist-plugin/user-role-editor": "^4.55",
    "wpackagist-plugin/wordpress-seo": "^14.0",
    "wpackagist-plugin/bulk-delete": "^6.0.2"
},
```

To save time you can create a composer_dev.json, composer_stage.json, and composer.json. Each time copy the content of _dev or _stage to composer.json before running the command `composer update --no-dev` and pushing to the project repo

If the process of generating, pushing, and merging the composer.lock file is done properly, you'll receive a ***JENKINS email notification****.* ***Jenkins*** *is a continuous integration and continuous delivery process that builds and manages the state of the website. In this case, Jenkins relies on composer.lock's update to run the processes set up by Campbell's team on their server, (e.g compiling the* ***LESS*** *files to* ***CSS****).*