CONFIGURATION AS DEPENDENCY

Managing Drupal 8 Configuration with Git and Composer
ERICH BEYRENT

➤ Senior Drupal Developer at BioRAFT

➤ Working with Drupal since 2004

➤ Drupal: https://drupal.org/u/ebeyrent

➤ Twitter: https://twitter.com/ebeyrent

➤ Github: https://github.com/ebeyrent
Moshe Weitzman
Moshe has been a consistent contributor to Drupal core and Contrib since November 2001. As such, he has pretty much touched the whole core code. He maintains the user.module, the bootstrap code, and the groups.drupal.org web site.

Justin Ludwig
Justin works for BioRAFT as a Development Engineer and am responsible for anything that gets thrown my way. He thrives in areas related to user management including SSO & permissions templates as well as adding some whizz-bang to sites with jQuery. Drupal has been his tool of choice since 2007.
AGENDA

➤ Overview of Drupal 8 configuration management
AGENDA

➤ Overview of Drupal 8 configuration management
➤ Managing your Git repository
AGENDA

➤ Overview of Drupal 8 configuration management
➤ Managing your Git repository
➤ Using Composer and Drupal Console
AGENDA

➤ Overview of Drupal 8 configuration management
➤ Managing your Git repository
➤ Using Composer and Drupal Console
➤ Multisite considerations
<configuration>
BACKGROUND

- Pre-Drupal 8 Challenges
  - Configuration cannot easily be represented in code
  - Configuration changes performed by clients and/or client services cannot be represented in code without going through patch/release with developers
BACKGROUND

➤ Drupal 8 Goals

➤ Allow non-developers to change configuration safely

➤ Clients can safely change configurations to the extent that they are allowed

➤ Provide an audit trail of all configuration changes made

➤ Easily and automatically create new instances of sites with default configurations
Two Types of Configuration in Drupal 8
DRUPAL 8 CONFIGURATION SYSTEM

- Standardizes configuration export into YAML files
- Used for both simple config and config entities (fields, views, workflows, roles, etc)
"Sites own their configuration, not modules."

- Alex Pott

https://www.chapterthree.com/blog/principles-of-configuration-management-part-two
DRUPAL 8 CONFIGURATION SYSTEM

- Contrib/custom modules have the same tools for enabling default configurations
- Default YAML files are stored with the module
- Once a module is installed, the site now owns the config and those YAML files are copied to the site-wide config directory
DRUPAL 8 CONFIGURATION SYSTEM

➤ Robust system for managing hard and soft/optional dependencies

➤ Nested module dependencies are automatically added to the configuration (when exported)
USEFUL CONFIGURATION MODULES

➤ Configuration Update
  https://www.drupal.org/project/config_update

➤ Configuration Tools
  https://www.drupal.org/project/config_tools

➤ Configuration Synchronizer
  https://www.drupal.org/project/config_sync

➤ Configuration Development
  https://www.drupal.org/project/config_devel

➤ Configuration Log
  https://www.drupal.org/project/config_log
WANT MORE INFORMATION?

Watch Moshe Weitzman’s DrupalNights presentation about Drupal 8’s configuration management system!

http://drupalnights.org/events/2015/moshe-weitzman-d8-configuration-management
There is no “recommended” workflow.

-Matthew Tift

https://www.lullabot.com/articles/configuration-management-in-drupal-8-the-key-concepts
BRO
DO YOU EVEN GIT?
Create two repositories:

- Code
- Configuration
MY GOD

THERE'S TWO OF THEM
YES, THERE'S TWO OF THEM

➤ Use git to manage individual site differences and conflict resolution

➤ Default configurations for all sites can be managed on the master branch of the configuration repository

➤ Each client site has its own branch of the Config Repo

➤ Code repository contains only custom modules/themes/libraries, “sites” directory and build scripts
GIT REPO STRATEGY: CODE REPO

➤ All dependencies are managed by Composer

➤ Package and version added to composer.json and it is built with all dependency dependencies resolved automatically
Configuration is just another dependency.

-Erich Beyrent
CONFIGURATION AS DEPENDENCY

- Manage configuration as a dependency in composer.json
- Use tags to identify configuration versions
"repositories": [
{
  "type": "composer",
  "url": "https://packagist.drupal-composer.org"
},
{
  "type": "vcs",
  "url": "https://github.com/weitzman/multiplesite-config.git"
}]
"require": {
    "composer/installers": "~1.0.20",
    "drupal-composer/drupal-scaffold": "~2.0.1",
    "cweagans/composer-patches": "~1.0",
    "drupal/core": "~8.0",
    "drush/drush": "8.x-dev",
    "drupal/console": "~1.0",
    "multiplesite-config/alpha": "dev-alpha",
    "multiplesite-config/bravo": "dev-bravo"
},
"config": {
    "preferred-install": {
        "multiplesite-config/*": "source"
    }
}

ichs-MacBook-Pro:multiplesite erich$ composer install
DrupalProject\composer\ScriptHandler::checkComposerVersion
loading composer repositories with package information
dating dependencies (including require-dev)
- Installing composer/installers (v1.2.0)
  Downloading: 100%
- Installing drupal-composer/drupal-scaffold (2.1.0)
  Downloading: 100%
- Installing cweagans/composer-patches (1.5.0)
  Loading from cache
- Installing symfony/event-dispatcher (v2.8.11)
  Loading from cache
- Installing symfony/polyfill-mbstring (v1.2.0)
  Loading from cache
- Installing symfony/console (v2.8.11)
  Downloading: 100%
- Installing psr/log (1.0.0)
  Loading from cache
- Installing symfony/debug (v2.8.11)
  Downloading: 100%
- Installing consolidation/output-formatters (1.0.0)
  Downloading: 100%
- Installing symfony/finder (v2.8.11)
  Downloading: 100%
- Installing webmozart/assert (1.1.0)
  Downloading: 100%
"extra": {
    "installer-paths": {
        "config/{$name}": ["type:bonefish-package"],
        "web/core": ["type:drupal-core"],
        "web/modules/contrib/{$name}": ["type:drupal-module"],
        "web/profiles/contrib/{$name}": ["type:drupal-profile"],
        "web/themes/contrib/{$name}": ["type:drupal-theme"],
        "drush/contrib/{$name}": ["type:drupal-drush"]
    }
},
"scripts": {
    "drupal-scaffold": "DrupalComposer\DrupalScaffold\Plugin::scaffold",
    "pre-install-cmd": [
        "DrupalProject\composer\ScriptHandler::checkComposerVersion"
    ],
    "pre-update-cmd": [
        "DrupalProject\composer\ScriptHandler::checkComposerVersion"
    ],
    "post-install-cmd": [
        "DrupalProject\composer\ScriptHandler::createRequiredFiles"
    ],
    "post-update-cmd": [
        "DrupalProject\composer\ScriptHandler::createRequiredFiles"
    ]
},
GIT REPO STRATEGY: CONFIG REPO

- Master branch contains the default configurations for any/all modules
- Each client site gets its own branch in the config repository
- For a site to work, we need both repos
- Master branch containing site defaults is mirrored from the Config Repo into the Drupal Repo (read-only) using git subtree split
DEV EP LER  W ORKFLOW

➤ Create a branch of Code Repo for a new ticket/card/bug/feature and do some work

➤ The Config Repo will be available to edit configs for sites where module is already enabled

➤ The same YAML will also have to be added to the module for sites where it was never enabled
CONFIG REPO BENEFITS

➤ While a feature is being worked on, a specific version of config can be used that won’t conflict with other features being developed

➤ If a problem arises with configuration, it can easily be “rolled back” by specifying the working version

➤ Exact replica of an entire site’s configuration can be built from scratch during development or debugging without any client data being used
MORE CONFIG REPO BENEFITS

➤ Helps determine if a bug involving config is actually solved rather than just solved for a particular piece of data
➤ Helps with testing unique configuration considerations
➤ Potential to assist with new site rollouts
WHAT ELSE?

➤ Theme repositories?
MULTISITE CONFIGURATION REQUIREMENTS

- `<?php return count($sites) >= 2; ?>`
- Configuration defaults exist and may need to be updated
- Each site is configured uniquely and may override defaults as required
- Both simple config (e.g. site name) and complex config (“config entities”) may need to be overridden
- A config may need to be changed for a single site or a group of sites
- Conflict management varies from case to case
An experiment in how to handle the "multisite" pattern in Drupal 8. This is when you run dozens or hundreds of very similar sites. You want these sites to vary configuration in slight ways, but still easily push out config changes to all. Examples include one site for each member of Congress, or one site for each musician in our portfolio.
IMPLEMENTATION

multisite

master config

multisite-config

master config

client a branch

client b branch

client c branch

*git subtree-split*
IMPLEMENTATION ALTERNATIVE

multisite

multisite-config

master config

client a branch

client b branch

client c branch
IMPLEMENTATION

master config

master config changes

client a branch

client b branch

client c branch

client c config changes

git merge
PUSHING COMMITS

➤ When fixing bugs while using a client site, a developer can choose to push commits to master config or to client config as needed. Pushing to client config happens automatically since that’s “origin”. If dev wants to integrate changes into multiplesite, add a remote pointing to multiplesite and then push commits there.

```bash
  git remote add multiplesite https://github.com/weitzman/multiplesite.git
  git checkout -b multiplesite-master multiplesite/master
  git cherry-pick <COMMITS>
  git push
```
IMPLEMENTATION

- master config
- client c branch
- config changes
- client c configuration
- client c production database
Multisite projects have unique config requirements
The proposed workflow should be able to handle all of those requirements elegantly
The process can be automated enough to reduce cognitive strain and allow everyone to focus their energy on what matters: the product!
RESOURCES

➤ https://www.drupal.org/developing/api/8/configuration

➤ https://dev.acquia.com/blog/ultimate-guide-drupal-8-episode-6-new-back-end-features-drupal-8

➤ https://github.com/weitzman/multiplesite

➤ https://www.lullabot.com/articles/configuration-management-in-drupal-8-the-key-concepts

➤ http://drupalnights.org/events/2015/moshe-weitzman-d8-configuration-management
BIORAFT IS HIRING!

http://bioraft.com/company/careers