



Configuration Workflow in Drupal 8

Mike Madison

About Me

Technical Architect @ Acquia

Organizer of Drupal GovCon
Maintainer of BLT + COD

D.O: mikemadison

Github: mikemadison13

LinkedIn: mikemadison

Twitter: mikemadison



Agenda

- Basics of D8 Configuration
- Managing Configuration Across Environments
- Config Split
- Config Ignore
- Configuration in Code

Topics Covered

- Managing different configuration for different environments (config split)
- Ignoring configuration entirely (config ignore)
- Basic configuration workflow (including git, CI, testing, etc.)
- Interaction with DevOps (using tools like Acquia BLT)
- Dependency interactions with other config, modules, and composer
- Working with configuration in code

Disclaimer

Recommended approach as of August 2018.
Subject to change.

<https://www.drupal.org/project/cmi2>

<https://www.drupal.org/project/drupal/issues/1613424>

<https://www.drupal.org/project/drupal/issues/2982056>

<https://www.phase2technology.com/blog/drupal-8-configuration-management>

Take Away #1

Keep up to date with all the happenings.



What is Configuration?

Configuration is the collection of admin settings that determine how the site default functions, as opposed to the content of the site.

Configuration will typically include things such as the site name, the content types and fields, taxonomy vocabularies, views and so on.

<https://www.drupal.org/docs/8/configuration-management>

What is Configuration?

Adding a Menu: “Config”

Adding a link to a Menu: “Content”

Adding a User Role: “Config”

Assigning a Permission to a Role: “Config”

Assigning a User to a Role: “Content”

What is Configuration?

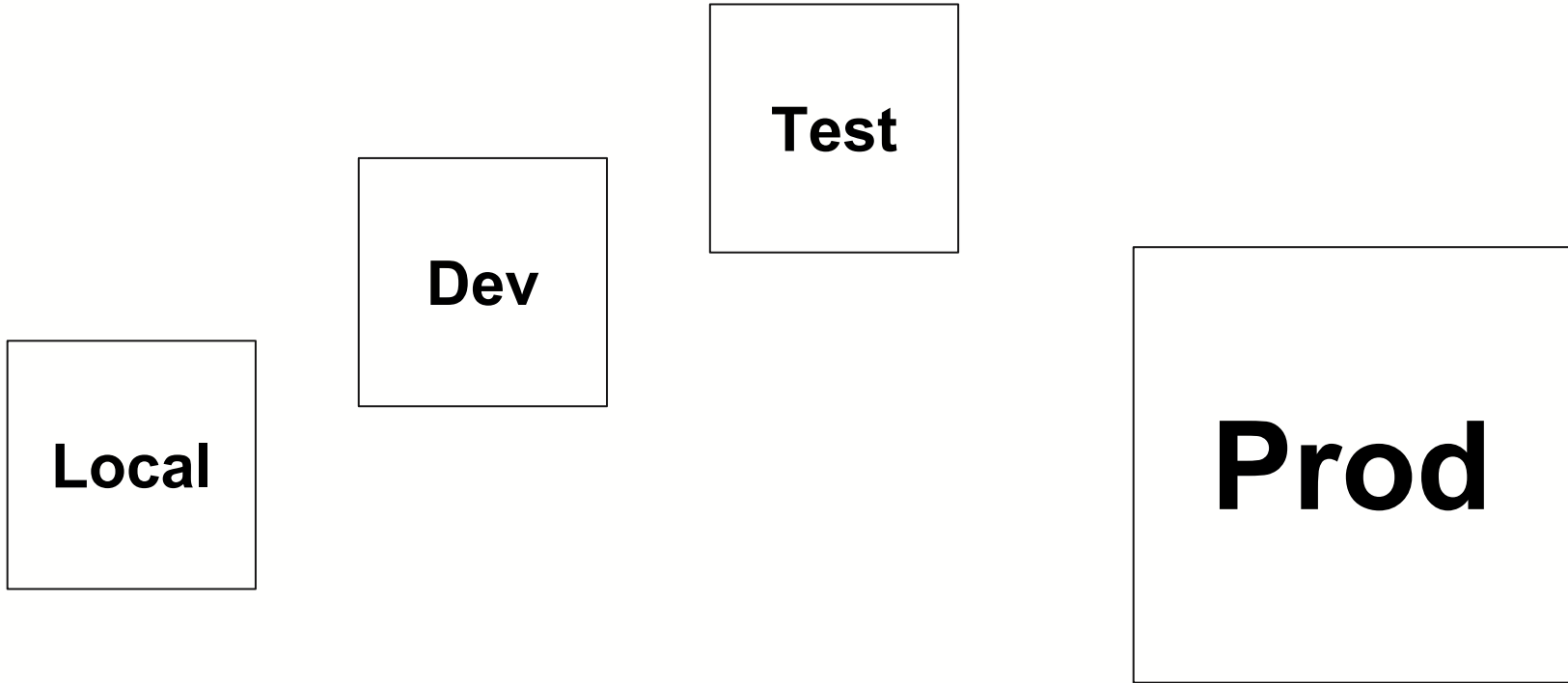
File System

- YML files

Database

- Config table

What is Configuration?



EXAMPLE

<https://github.com/Drupal4Gov/Drupal-GovCon-2017/>

- What is configuration?
- How is it stored?
- How do you read it?
- How do you change it?

Problem

If it's in the database...

- How to execute a proper workflow?
- How to preserve changes after updates?
- How to not break your @#%!?

Example: Configuration Workflow

Development

1. Install Drupal
2. Create Blog Content Type
3. Add fields and configure form / display
4. Add Blog View
5. Export Configuration
6. Commit yml files
7. Writes Behat Tests
8. Pull Request

Example: Configuration Workflow

Code Review / QA

1. Install Drupal
2. Import Configuration
3. Configuration Creates Content Type, Fields, Display, View, etc.
4. Review Blog Content Type
5. Review View
6. Review Tests
7. Merge Pull request

Example: Configuration Workflow

Deployment

1. Import Configuration
2. Blog Features on Production

Take Away #2

Config will bite you.
Limit where it can bite you.



Benefits of Configuration Workflow

- Develop new features for a site outside of production
- Mirror environments without having to move databases
- Bootstrap and test Drupal during CI
- Store in version control

Challenges of Configuration Workflow

- Core Config is all or nothing
- Core does not import configuration by default during installs or database updates
- No implicit / explicit logical, conditional, or environment specific capabilities
- Drastic changes (e.g. deleting a field) may require data manipulation and/or custom development

Overcoming Challenges

Core Config is all or nothing

Solutions:

Store all active configuration in repository
(e.g. config/default)

Don't **ever** make config changes on a production site. Always rely on workflow.

Use contrib to limit what core manages.

Overcoming Challenges

Core does not import configuration by default during installs or database updates

Solutions:

Use automation tools (e.g. Acquia BLT) that manage configuration for you

Host with a company that provides deployment hooks easy CI integration

Overcoming Challenges

No implicit / explicit logical, conditional, or environment specific capabilities

Solutions:

Expand Core features with contrib modules such as Config Split / Config Ignore

Use automation tools (e.g. Acquia BLT) to detect environments and activate specific version of config / config splits

Overcoming Challenges

Drastic changes (e.g. deleting a field) may require data manipulation and/or custom development

Solutions:

Limit data model changes to production sites (BEST PRACTICE)

Write migrations prior to removing key features (BEST PRACTICE)

Solutions: Store Config

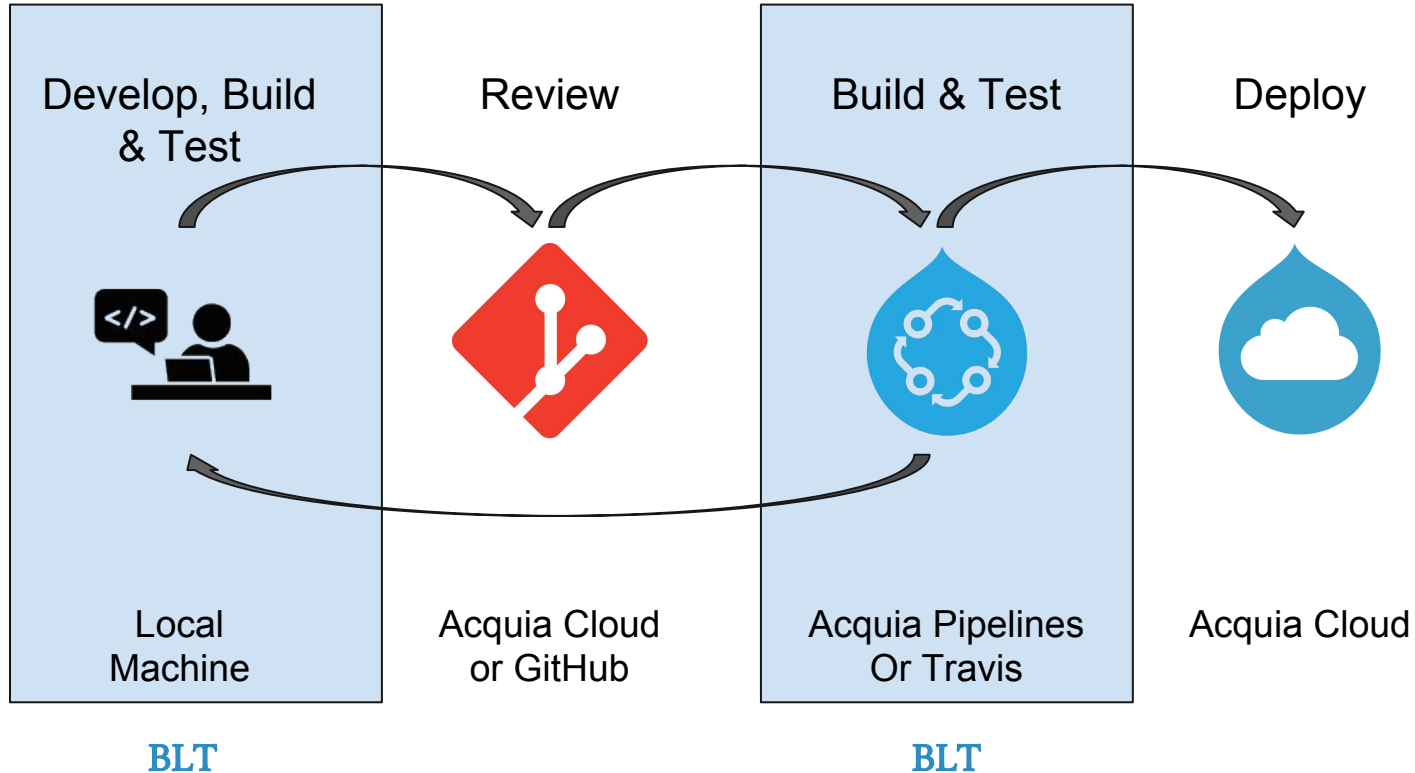
<https://github.com/Drupal4Gov/Drupal-GovCon-2017>

Solutions: Automation with BLT

- An Acquia-built tool for creating *new projects* from a **standardized template**
- A common set of tools for building, testing, validating, deploying, etc.
- A collection of **commands for automating** common project tasks
- A repository and enforcer of **best practices**



Solutions: Automation with BLT



Take Away #3

There is no try. Do or do not.



EXAMPLE

Managing Block Config

- Drush
- Git
- BLT

EXAMPLE

Managing Config Splits

- Why config split?
- What can you do with it?

EXAMPLE

Managing Config Ignores

- Why config ignore?
- What can you do with it?

EXAMPLE

Changing / Using Config in Code

<https://github.com/mikemadison13/govcon2018php>

Take Away #4

Core config needs help. BUT there is good help in contrib.



Questions?