Cornell DrupalCamp VI
camp.drupal.cornell.edu

September 26-27, 2019
Cornell University
Ithaca, NY

be cool, and

# Commit! Your! Config! -- A Primer

Alison Jo McCauley
Drupal Developer, Cornell University

Comin' atcha

- ♥ What is Configuration Management ("CMI")
- ♥ How to use CMI
- ♥ Using CMI with Dev > (Test >) Live
- ♥ Configuration in code
- ♥ Pain points

- ♥ Still not sure?!?! -- Why you should **Commit! Your! Config!**
- ♥ Modules
- ♥ Brief mention of D8 migrations
- ♥ Questions? Lingering concerns?

- ♥ BONUS: Your next steps
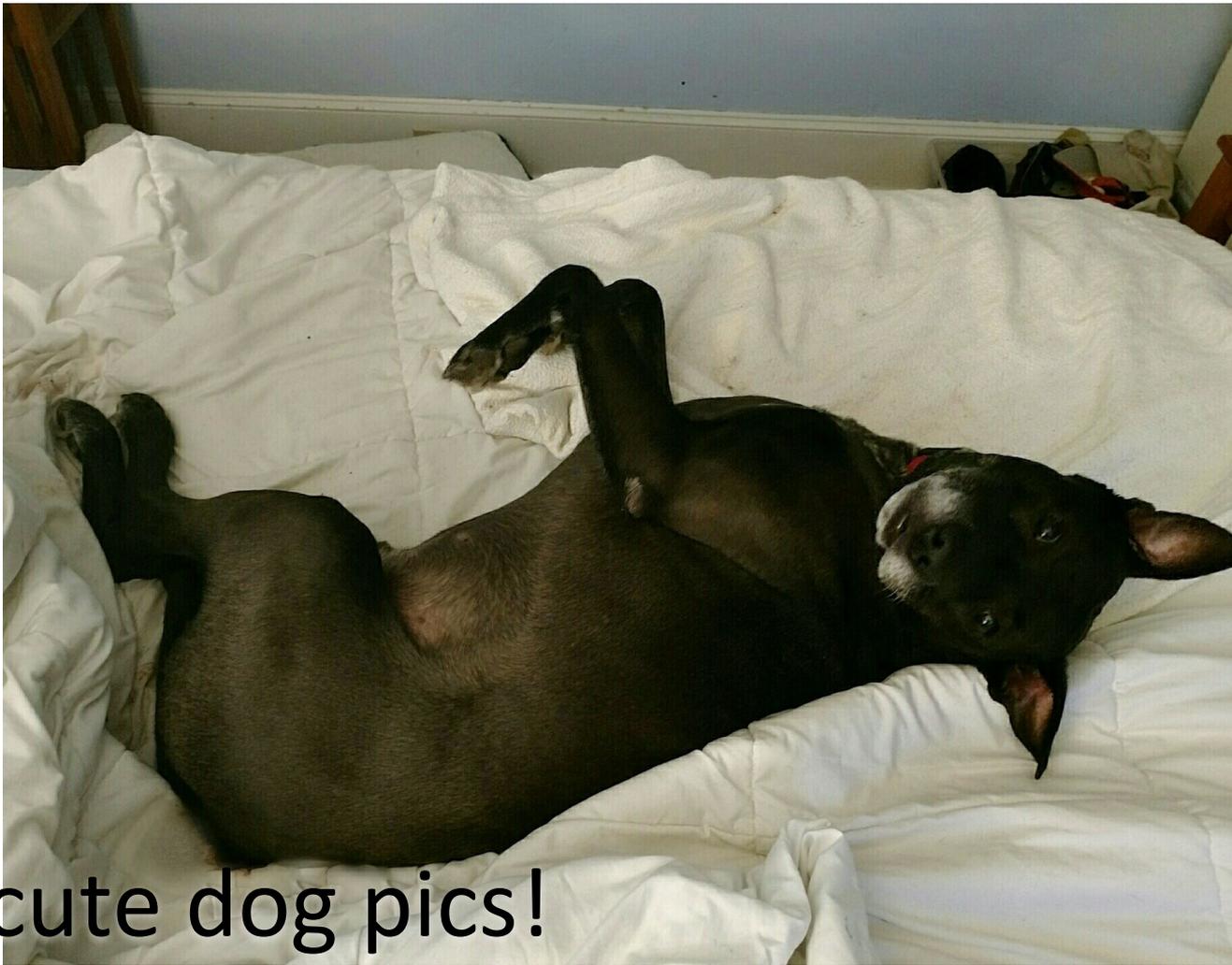
...and cute dog pics!

What is Configuration Management ("CMI")

PSYCH!

First of all, have you used...

♫ Drupal 8?

♫ configuration management in Drupal 8?

♫ Features in Drupal 7?

▶ *Have you used "configuration management" in Drupal 7?*

♫ Think about the tools ^^ you use or want to be using.

♫ Think about your wish list, when you came into this session.

What is Configuration Management ("CMI")

What is Configuration Management ("CMI")

🎵 ALL site settings go in the `config` table of the database.

What is Configuration Management ("CMI")

🎵 ALL site settings go in the `config` table of the database.

Obviously that's a lie.

Don't worry, plenty of site settings are confusingly *not* in the config table.

What is Configuration Management ("CMI")

🎵 ALL site settings go in the `config` table of the database.

🎵 Consistency

🎵 Features in Drupal 7, "ish" (but simpler)

What is Configuration Management ("CMI")

♫ ALL site settings go in the `config` table of the database.

♫ Consistency

♫ Features in Drupal 7, "ish" (but simpler)

Learn more:

♫ [History of the Configuration Management Initiative](#)

♫ [CMI 2.0 discussions](#) -- part of effort to make Drupal easier to use

#goals

What is Configuration Management ("CMI")

🎵 ALL site settings go in the `config` table of the database.

🎵 Consistency

🎵 Features in Drupal 7, "ish" (but simpler)

Learn more:

🎵 [History of the Configuration Management Initiative](#)

🎵 [CMI 2.0 discussions](#) -- part of effort to make Drupal easier to use

#goals

If you care, chime in!

OK but, what *is* configuration?

♫ or, what are configurations?

♫ or…?

♫ Silly you --

"ALL site settings go in the `config` table of the database." ← there ya go!

What?! ...and, How?!

What?!

🌞 Individual configurations -- examples:

```
core.extension
views.settings
views.view.events
```

(What?! ...and How?!)

☀ Config:

▶ Entity types (+ bundles): node types, vocabs, custom block types, media types…

▶ Modules enabled

▶ Views

▶ User roles and perms

▶ Misc site settings (including those from custom modules)

▶ Each menu setup

▶ Non-custom block settings

▶ Block layout settings

☀ Not config:

▶ Entities -- nodes, terms, users, files…

▶ Menu links

▶ Custom blocks

(What?! ...and How?!)

☀ Config:
- ▶ Entity types (+ bundles): node types, vocabs, custom block types, media types…
- ▶ Modules enabled
- ▶ Views
- ▶ User roles and perms
- ▶ Misc site settings (including those from custom modules)
- ▶ Each menu setup
- ▶ Non-custom block settings ← *trickiness*
- ▶ Block layout settings ← *warning*

☀ Not config:
- ▶ Entities -- nodes, terms, users, files…
- ▶ Menu links *trickiness*
- ▶ Custom blocks ← *warning*

What?! ...and, How?!

☀ Configuration is tied to one site's unique `hash_salt` (in settings.php).

How?! -- to work with config

☀️ "Single item" import/export

☀️ "Full archive" import/export

☀️ Config editor

☀️ Drush -- available commands:
   `drush | grep config-`

How?! -- to work with config

BUT WAIT! THERE'S MORE! *

☀ "Import all"

☀ "New," "Changed," "Removed"

☀ Active (current setting) and Staged (to be imported)

* if you commit your config 😛

Flashback!

What is Configuration Management ("CMI")

🎵 ALL site settings go in the `config` table of the database.

🎵 Site settings are stored in a consistent manner.

🎵 Features in Drupal 7, "ish" (but simpler).

^^ what??

...similar "ish" to Features in Drupal 7

CMI:

Synchronize ALL^{LOL} site settings between **identical** sites.

i.e. between different environments of the same project

I-dentical.

...similar "ish" to Features in Drupal 7

CMI:

Synchronize ALL^{LOL} site settings between **identical** sites.

i.e. between different environments of the same project

(whereas)

Features:

Organize related config and/or build reusable functionality.

How?! -- to use CMI with Dev > (Test >) Live

How?! -- to use CMI with Dev > (Test >) Live

Use CMI with Dev > (Test >) Live

What's your process for changing or creating functionality on your Drupal site?

How?! -- to use CMI with Dev > (Test >) Live

- Synchronize between environments
- Mess with stuff (..."develop"...) in your Dev environment
- (Test on test/staging environment…)
- THEN, go live!

How?! -- Site config in your code

How?! -- Site config in your code

Reminder! Drupal stores ***active*** site settings in the `config` DB table.
...BUT / AND!

☂ yaml (`.yml`)
- ▶ (like so many things in Drupal 8!)

☂ Where?
- ▶ "Usually" sites/default/files/sync
- ▶ OR! (if you can) outside your project root
- ▶ Specify in settings.php:
  `$settings['config_sync_directory']`
  - ■ AND check on your hosting provider directory structure

How?! -- Site config in your code

Reminder! Drupal stores ***active*** site settings in the `config` DB table.
...BUT / AND!

☂ yaml (`.yml`)
    ▶ (like so many things in Drupal 8!)

☂ Config files are... those configurations, in yaml!
☂ Remember those config names? Now as files--

```
core.extension.yml
views.settings.yml
views.view.events.yml
```

Flashback!

How?! -- to use CMI with Dev > (Test >) Live

↗ Synchronize between environments

↗ Mess with stuff (..."develop"...) in your Dev environment

↗ (Test on test/staging environment…)

↗ THEN, go live!

↗ NOW with config that's in your codebase! 🎉

Pain points?

What pain points?

lolololollolololololol

#laughsob
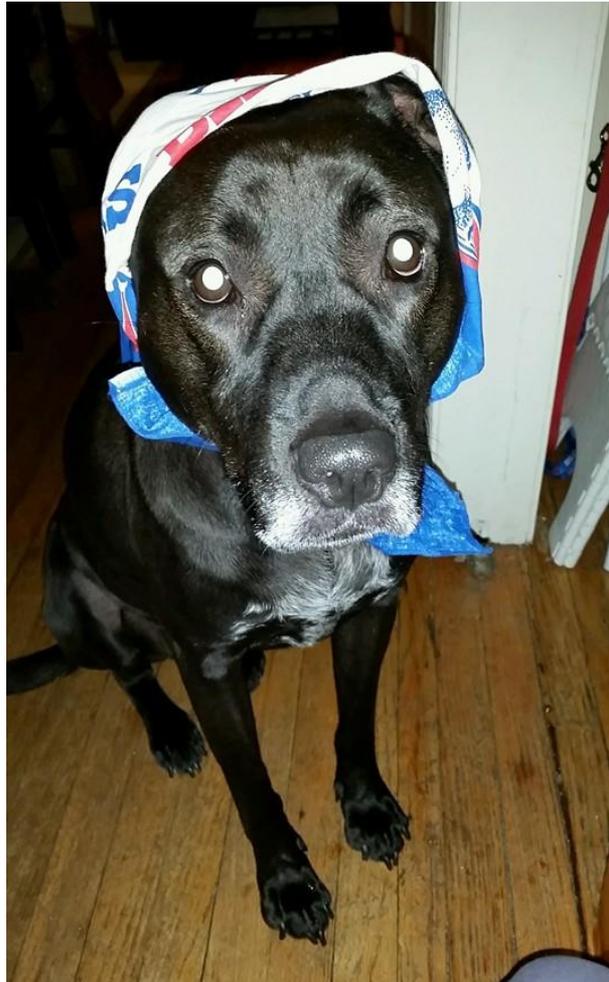
Pain points

☀ "Import all" has no confirmation screen -- what!?

Pain points

- Learning the errors / warnings
  - The following items in your active configuration have changes since the last import that may be lost on the next import.
  - ...your thing depends on *cute.puppies.settings* which does not exist...
  - ...cannot delete *fields.field.field_dog_rating* because there's data…
  - UUID mismatches (not an error, just annoying)
  - Blocks (ugh)

- Learning names of configs
- Everything at once, OR, tiny pieces one at a time
- New/changed configs due to updates
- Pain specific to committing your config…? Not really.
  - Simplicity, compared to Features

Why you should **Commit! Your! Config!**

Why you should **Commit! Your! Config!**

☃ Version control.

☃ VERSION CONTROL!!

☃ V E R S I O N   C O N T R O L

☃ Mini-flashback:

Everything at once, OR, tiny pieces one at a time

^^ UNLESS you **Commit! Your! Config!**

yayyy!!

(If time allows)

Modules

⚡ **[Configuration Manager](#) in Drupal 8 core**

⚡ [Config ignore](#) -- completely ignore certain configurations (examples)

⚡ [Configuration split](#) -- different settings per environment (examples)

Also:

⚡ [Configuration installer](#)

⚡ [Configurable Blocks](#)

⚡ [Configuration Update Manager](#) -- not what I thought it was going to be

⚡ [D8 Features](#)

Modules

⚡ **[Configuration Manager](#) in Drupal 8 core**

⚡ [Config ignore](#) -- completely ignore certain configurations (examples)

⚡ [Configuration split](#) -- different settings per environment (examples)

Also:

⚡ [Configuration installer](#)

⚡ [Configurable Blocks](#)

⚡ [Configuration Update Manager](#) -- not what I thought it was going to be

⚡ [D8 Features](#)

Modules

⚡ **[Configuration Manager](#) in Drupal 8 core**

⚡ [Config ignore](#) -- completely ignore certain configurations (examples)

⚡ [Configuration split](#) -- different settings per environment (examples)

Also:

⚡ [Configuration installer](#)

⚡ [Configurable Blocks](#)

⚡ [Configuration Update Manager](#) -- not what I thought it was going to be

⚡ [D8 Features](#)

(If time allows)

Migrating to Drupal 8 (from D6/D7)

...it depends! 🎉

- Content types, vocabularies (taxonomy), …

- Look in module code! (core/contrib)

- dblog settings example:
  https://www.drupal.org/docs/8/api/migrate-api/migrate-destination-plugins-examples/migrating-configuration

(If time allows, mention composer re: "installing"/"enabling" things…)
- composer require + core.extension
-

Now that I've convinced you...

❄️ Questions?

❄️ Lingering concerns?

Also, links!

🐾 [D.o: Managing your site's configuration](#)

🐾 [History of the Configuration Management Initiative](#)

🐾 [CMI 2.0 discussions](#) -- part of effort to make Drupal easier to use *#goals*

If you care, chime in! Don't let the convo only be between "core developers."

@alisonjo2786 everywhere; alison@cornell.edu

Your next steps...

1. Enable "configuration manager" (core module)
2. Export your site's configuration (admin > configuration > development > configuration synchronization > export > download the whole archive)
3. Extract those files to a location in your repo that makes sense for you (see slide 32)
4. Update settings.php with where you put those config files (see slide 32)
5. Commit those files to your repo
6. 🎉 PARTY 🎉