MOUNTAIN CAMP

zehnplus.

MANAGING DRUPAL

WITH COMPOSER

AND GIT

AND DRUSH

Bill Seremetis (bserem) - Drupal Implementor
Vasilis Papoutsakis (Zekvyrin) - Back-End Developer

MOUNTAIN CAMP

zehnplus.

WAYS TO MAINTAIN A DRUPAL PROJECT

BAD WAY

STILL USABLE TODAY

1. Get the zip file
2. Extract it and upload with ftp
3. Install
4. Get a module/theme zip file
5. Go to step #1. Repeat.

I just hope it is SFTP

OK WAY

1. Use SSH instead of FTP.
2. Install Drush, get Drupal code with `drush dl`

Well... at least you use SSH.

OK IMPROVED

1. Add **everything** in a Git repo.

Hurray! You have an ugly git history, but you have a history!

MOUNTAIN CAMP

zehnplus.

REALLY NICE WAY

VALID UNTIL D7

Use SSH+Drush and manage your **custom code** with Git:

1. Ignore all Drupal Core+Contrib code
2. Use a **drush makefile**
3. Do whatever you think of about third-party libraries... (ssh and unzip and track with git)

Congratulations! This is probably the best way to manage a D6/D7 site. It is good, it uses all best practices and it works. Period.

MOUNTAIN CAMP

zehnplus.

ENTER `COMPOSER`

DRUPAL IS OFF THE ISLAND!

COMPOSER

WHAT EVERYBODY ELSE WAS USING FOR YEARS

- Package manager for PHP
- For packages! Not extensions
- Can handle dependencies of third-party packages

Drupal acted smart and adopted composer in D8.
It was, mostly, a good thing...

HOW TO USE COMPOSER

1. Install composer (https://getcomposer.org/)
2. Create your Drupal codebase using drupal-composer/drupal-project
3. Add contrib modules/themes
4. Do your Drupal work

Note: the above method is not suitable for core development.

MOUNTAIN CAMP

zehnplus.

COMPOSER-PROJECT

- Github: `https://github.com/drupal-composer/drupal-project`
- Recommended by drupal.org
- Downloads Drupal code + dependencies
- Also downloads Drupal Console & Drush locally
- Manages patches

```
composer create-project drupal-composer/drupal-project:8.x-dev <directory> --stability dev --no-interaction
```

MOUNTAIN CAMP

zehnplus.

STARTING A NEW PROJECT

STEPS TO BE DONE BY THE FIRST DEVELOPER

- Download and install Drupal (`drush site:install`)
- Export configuration (`drush config:export`)
- Commit and push to Git repository

Other developers can now join (clone and get db)

INSTALL FROM AN EXISTING CONFIG (D8.6+)

```
$ drush site:install --existing-config
```

- Requires Drupal 8.6+ and Drush 9.4+
- Works for specific profiles like minimal (not standard yet). Work in progress!
- Alternatives: config_installer module, dump & share initial database

ADDING A MODULE

```
composer require drupal/<modulename>
composer require drupal/<modulename>:<version>
```

ADDING A DEV-ONLY MODULE

```
composer require --dev drupal/devel
```

Use this with `config split` or CMI2 once it is released

IDENTIFY OUTDATED CODE

```
composer outdated drupal/*
```

What used to be `drush ups`

MOUNTAIN CAMP

zehnplus.

UPDATE CORE

```
composer update drupal/core webflo/drupal-core-require-dev --with-dependencies
```

Don't ask why... just do it :)

UPDATE A MODULE

```
composer update --with-dependencies
drupal/<modulename>
```

REMOVE A MODULE

```
composer remove drupal/<modulename>
```

If the module was added with `--dev` flag, you might get a prompt to confirm the removal from require-dev

PATCHES

Composer project comes with composer-patches

```
"extra": {
  "patches": {
    "drupal/pdf": {
      "library adjust weight": "https://www.drupal.org/files/issues/2019-01-09/3024877-2-adjust-
    },
  }
}
```

LOCK HASH WARNINGS

well.. sometimes hash on composer.lock isn't correct and will throw warnings.

```
composer update --lock
```

(will just update hash and nothing else)

GIT

Use the default `.gitignore` provided by drupal-project.

WHAT SHOULD BE COMMITED

- composer.json and composer.lock
- configuration folder (def: `./config`)
- any custom code: `./web/modules/custom` `./web/themes/custom`

WHAT SHOULDN'T BE COMMITED

- Vendor folder `./vendor`
- Drupal Core `./web/core`
- Contrib modules (`./web/modules/contrib`) or themes (`./web/themes/contrib`)

COMMON PITFALLS

- Always `import` first and then work and export
- Avoid installing dev-dependencies on production
- Do not run `composer update`

DEPLOY TO PRODUCTION

Don't run Composer require or update on the production server!

- Ideally rsync/copy files from somewhere else.
- If you can't do otherwiser, run with warm caches.
- Lock the DB!

WORTH KNOWING

- Drupal Composer Project comes with drush 9 and drupal console by default.
- Drush 9 is installed on a per-project level. Not a system level.
- You need to add `drush-launcher` in your binaries so as to locate and use the proper drush.
- Most third-party php libraries will be downloaded automatically
- For non-php assets you can use `https://asset-packagist.org/`

DEV ONLY CONFIG

Common approach: `config_split` allows you to split some configuration and enable/disable them per enviroment
For example having devel active only on dev, or reroute_email activated anywhere but on production

```
$config['config_split.config_split.dev']['status'] = TRUE;
```

Alternative:

```
drush cex --skip-modules=devel,reroute_email
drush cim --skip-modules=devel,reroute_email
```

CONFIGURATION SYNC

- Export config with `drush cex`
- Add all files or selectively (ex: the view you just worked)
- Commit
- Import on other instance with `drush cim`

STRUCTURE SYNC

Allows to export/import taxonomies, blocks, menu links as configuration!

```
drush eb/ib (blocks)
drush et/it (taxonomies)
drush em/im (menu links)
```

.ENV FILES

```
MYSQL_DATABASE=’db_name’
MYSQL_HOSTNAME=’localhost’
MYSQL_PASSWORD=’secret’
MYSQL_PORT=’3306'
MYSQL_USER=’db_user’
```

Change settings.php:

```
$databases[‘default’][‘default’] = [
 ‘database’ => getenv(‘MYSQL_DATABASE’),
 …
 …
 …
```

zehnplus.

THANKS!

Bill Seremetis (bserem)

Vasilis Papoutsakis (Zekvyrin)

zehnplus.ch