Building a CI pipeline with BLT and Docker

Florida Drupal Camp 2022
Matthew Ramir (@crasx)
matthew.ramir@(bounteous|gmail).com

bounteous

Matthew Ramir
@crasx
they/them

bounteous

A little about me

- Chicago -> Denver
- BS in CS.
- Started playing with Drupal in 2012
  - First Drupal 7 site website was for the LGBTQ+ group in college
- Backend developer - Clients heavily in commerce/finance.
- Disabled (Cerebral Palsy)


bounteous

bounteous

AGENDA

Problem Statement

Some assumptions

Creating a Build Environment

Docker Setup

Tooling

Composer, phpcs, BLT, and other goodies.

Acquia BLT

Demystifying Build and Launch tool

Continuously Integrating

Bringing it all together in a pipeline

What are we solving for?

- Multiple developers modifying the codebase
- Ease the review process burden.
- Free up developers to write code instead of deploy it.

A Little Scope

- Out of scope:
  - Automated Testing
  - Deployments
  - SSH key management
- Using bitbucket, gitlab, other ci tool that can build Docker

How Are We Solving The Problem?

- Custom docker image
- Docker compose as an orchestrator
- Composer/BLT/PHP tools for validation and integration

Build Environment

Docker compose

Our docker file takes lots of arguments.

- Software versions
- CI info (branch name)
- SSH Keys
- API Keys

Docker Compose helps us organize them.

```
services:
 drupal:
   build:
     context: .
     dockerfile: scripts/docker/Dockerfile-drupal
     args:
       - PHP_VERSION
       - NODE_VERSION
       - SSH_KEYS
   environment:
     - BITBUCKET_BRANCH
     - BITBUCKET_TAG
     - ACSF_API_USERNAME
```

Provisioning the Build Environment

Core packages needed

- PHP
- NPM/NVM
- Openssh-client

```
# scripts/docker/Dockerfile-drupal

ARG PHP_VERSION
FROM php:${PHP_VERSION}-alpine
RUN apk add --no-cache --update   unzip git curl rsync wget
openssh-client … composer
RUN docker-php-ext-install -j$(nproc) gd opcache …


# Alternatively
FROM cimg/php:${PHP_VERSION}-node
ARG UID=3434
```

Provisioning the Build Environment

The Docker community provides some great starting points.

- cimg - Circle CI images
- Lando php images
- Docker4Drupal

Docker development environments on d.o

```
# scripts/docker/Dockerfile-drupal

ARG PHP_VERSION
FROM php:${PHP_VERSION}-alpine
RUN apk add --no-cache --update   unzip git curl rsync wget
openssh-client … composer
RUN docker-php-ext-install -j$(nproc) gd opcache …


# Alternatively
FROM cimg/php:${PHP_VERSION}-node
ARG UID=3434
```

Tooling

Most of our tooling can be installed via composer

- phpcs / phpcbf
- blt
- drush

```
# Gather project composer config
COPY composer.* /app/
COPY patches /app/patches

# Install dependencies
RUN composer install;

# Copy the rest of the code in.
COPY . /app
WORKDIR /app
ENTRYPOINT ["vendor/bin/blt", "--no-interaction",  "--ansi",
"--verbose", "--environment=ci" ]
```

Application & Entrypoint

We copy in the rest of the custom code after composer install.

Our entry point is BLT with a couple of args to target CI.

```
# Gather project composer config
COPY composer.* /app/
COPY patches /app/patches

# Install dependencies
RUN composer install;

# Copy the rest of the code in.
COPY . /app
WORKDIR /app
ENTRYPOINT ["vendor/bin/blt", "--no-interaction",  "--ansi",
"--verbose", "--environment=ci" ]
```

Testing and Rapid Iteration

```
# Using docker allows for easy local debugging
docker-compose build drupal
docker build -f scripts/docker/Dockerfile-drupal --build-arg PHP_VERSION=8.0

docker-compose run drupal artifact:deploy
```

Acquia Build & Launch Tool

Acquia BLT

- CLI tool for CI/CD and Drupal development
- Not just for Acquia hosting!
- Uses a yaml config file (blt/blt.yml)
  - Configurable per environment
- Convenient “hooks” for frontend tasks

BLT Configuration

Not just for Acquia!

```
project.human_name: My awesome project

# true = dependencies will be built during deploy
# false = you should commit dependencies directly.
deploy.build-dependencies: true

# Define docroot/webroot, defaults to /docroot
docroot: ${repo.root}/web
deploy.docroot: ${deploy.dir}/web

git.remotes:
 # Pantheon/Platform.sh upstream (?)
 - ssh://user@foo.dev:2222/~/repositories.git
```

BLT is extensible

Easily define and extend blt commands

Thanks robo! - https://robo.li/

```php
namespace Foo\Blt\Plugin\Commands;
use Acquia\Blt\Robo\BltTasks;
class FmSyncCommands extends FmCommandBase {
 /**
  * @command foo:artifact:post-build
  * @description Validates that a site has been selected
  */
 public function postBuild() {
```

blt validate

- PHPCS - coding standards
- composer.lock validation
- phpstan?

Also - blt test

blt artifact:build

- Run frontend build tasks
- Copy custom files (composer, modules, themes, profiles) to build directory
- Composer install
- Sanitize
- Configure simplesaml
- Custom post build tasks

blt artifact:deploy

- Create Blank Directory
- Init git, add remotes, check out branch, merge
- Run artifact:build
- Commit changes
- Tag / Push

Continuously Integrating

-

Putting it in a pipeline

```
- validate:
    name: 'Validate code.'
    script:
      - *setup
      - docker-compose build --progress plain
      - docker-compose run drupal validate
    caches:
      - docker
```

Putting it in a pipeline

```
build-branch:
 - step: &build-branch
     name: 'Build custom branch.'
     script:
       - *setup
       - docker build drupal
       - docker run drupal artifact:deploy --commit-msg
"Bitbucket build - $BITBUCKET_BUILD_NUMBER, Commit
$BITBUCKET_COMMIT" --branch "${DESTINATION}"
     caches:
       - docker
       - common
```

Thank you.

Matthew Ramir
Sr. Web Developer

773-428-3678

@crasx
matthew.ramir@bounteous.com