

Build & Launch Tools (BLT)

Automating best practices for enterprise sites

Who are you?

- Matthew Grasmick
- @grasmash on Drupal.org, twitter, etc.
- Acquia Professional Services, 4yrs
- Drupalist, 9yrs
- Maintainer of BLT

Let's start with the problem

that we're solving.









FTP



This was good

It was fast and simple.

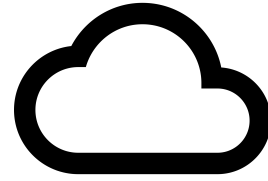
But it had problems!

- We'd **overwrite** each other's files
- We **couldn't track** or revert changes
- It **wasn't redundant** or resilient



FTP



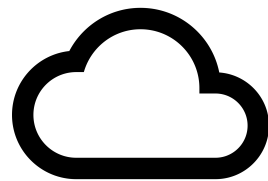


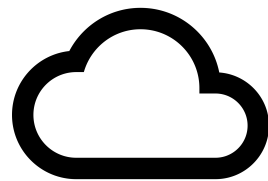
This was better

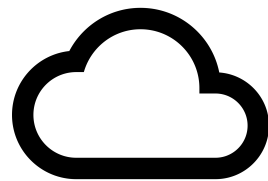
But also a bit more complicated.

Meanwhile...

Our **backend tools** were evolving





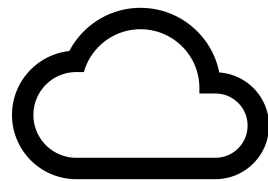


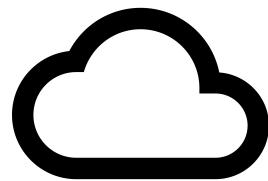
This was better

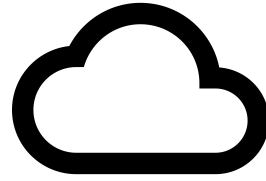
But also a bit more complicated.

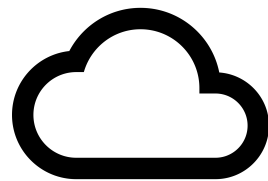
Meanwhile...

Our **front end tools** were changing too







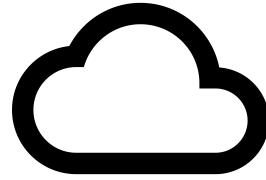


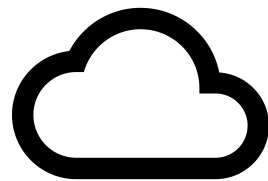
This was better

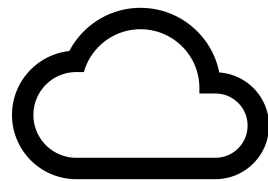
But also a bit more complicated.

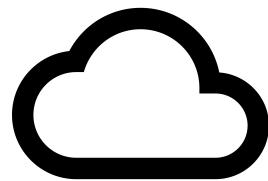
Meanwhile...

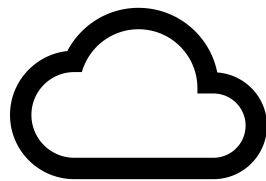
We starting getting great **testing tools**

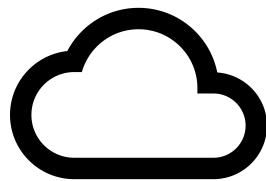










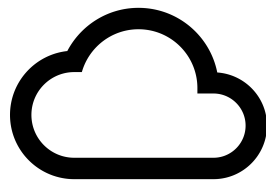


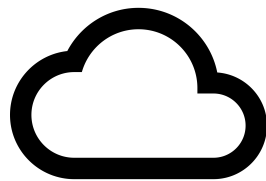
This was better

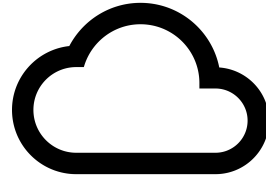
But also a bit more complicated.

Meanwhile...

Our **local environments** were changing





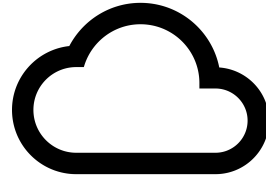


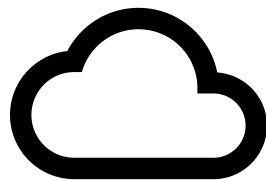
This was better

But also a bit more complicated.

Meanwhile...

The **PHP community** continued to evolve



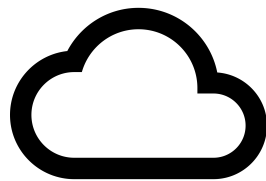


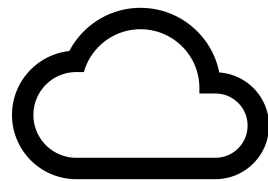
These are all great tools!

That we **run locally**.

Teams have many machines

And **continuous integration** requires even more machines





git

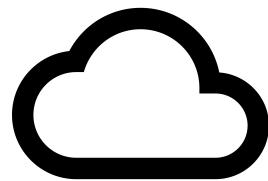


git



VAGRANT





git



At some point it became very complex

And **painful**.

This is the problem

We're trying to **solve**.

Real world consequences

1. You don't use these tools

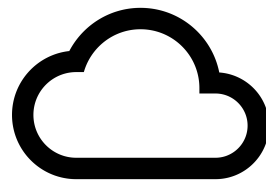
- a. You don't have a build process to manage dependencies
- b. You don't use automated testing
- c. You don't have a CI process

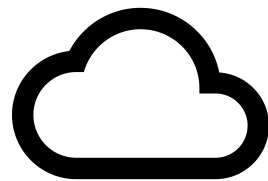
2. You spend a long time orchestrating these tools

- a. You painstakingly create your own build process, for every project
- b. You configure your own CI setup, for every project
- c. You maintain it constantly, for every project

There has to be a way...

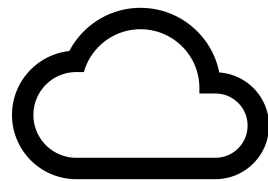
To approach this more consistently and efficiently.





git



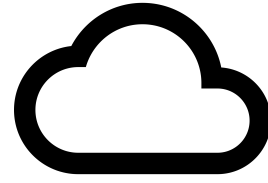


git



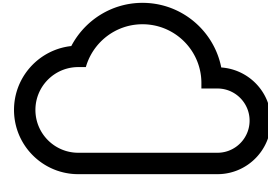
Build & Launch Tools

BLT



git





git



Package, integrate, automate

A stack of **Drupal** development tools.

What's in the box?














BLT high-level features:

- Open source
- Standardized template for Drupal 8 sites
- Set of tools for building, testing, deploying
- Commands for automating usage of those tools

Template

What is the template?

- Pre-defined structure for files and directories
- Boiler plate code and configuration

 grasmash BLT-123: This is a good commit.		Latest commit 0036916 a day ago
 config/default	BLT-123: This is a good commit.	3 minutes ago
 docroot	BLT-123: This is a good commit.	3 minutes ago
 drush	BLT-123: This is a good commit.	3 minutes ago
 factory-hooks/pre-settings-php	BLT-123: This is a good commit.	3 minutes ago
 files-private	BLT-123: This is a good commit.	3 minutes ago
 hooks	BLT-123: This is a good commit.	3 minutes ago
 patches	BLT-123: This is a good commit.	3 minutes ago
 tests	BLT-123: This is a good commit.	3 minutes ago
 .drush-use	BLT-123: This is a good commit.	3 minutes ago
 .editorconfig	BLT-123: This is a good commit.	3 minutes ago

Opinionated placement of...

- Settings files
- Exported configuration
- Modules, themes, profiles, etc.
- Drush commands
- Patches

Nothing earth-shattering.



Actually, it's kind of a big deal.

Why is it helpful?

- Faster on-boarding
- Better collaboration
- Simpler hand-offs

Tools

We're all using the same things

- The **same tools**, the same versions
 - Testing tools
 - PHP Packages
 - Modules
- Preconfigured to **work together**

Also a big deal.

What makes this powerful?

Automation!



Make complex things simple

- **Create** an entire project
- **Boot** (& create) a VM
- **Validate** all of your code
- **Test** you application
- **Sync** environments
- **Deploy** to cloud

... in one command

Seeing is believing.

Warning: we're going CLI.

Creating a new project

Use composer

and then wait a bit..

```
$ composer create-project  
acquia/blt-project  
[project-name]  
--no-interaction
```

You have a complete project!



This repository

Search

[Pull requests](#) [Issues](#) [Gist](#) **grasmash** / **blted8** Watch ▾

0

 Star


0

 Fork

0

[Code](#)[Issues](#) 0[Pull requests](#) 0[Projects](#) 0[Wiki](#)[Pulse](#)[Graphs](#)[Settings](#)

No description or website provided. — [Edit](#)

 1 commit 1 branch 0 releases 1 contributor

Branch: master ▾

[New pull request](#)[Create new file](#)[Upload files](#)[Find file](#)[Clone or download ▾](#)**grasmash** BLT-123: This is a good commit.

Latest commit 0036916 a day ago

 [config/default](#)

BLT-123: This is a good commit.

3 minutes ago

 [docroot](#)

BLT-123: This is a good commit.

3 minutes ago

 [drush](#)

BLT-123: This is a good commit.

3 minutes ago

 [factory-hooks/pre-settings-php](#)

BLT-123: This is a good commit.

3 minutes ago

 [files-private](#)

BLT-123: This is a good commit.

3 minutes ago

 [hooks](#)

BLT-123: This is a good commit.

3 minutes ago

 [patches](#)

BLT-123: This is a good commit.

3 minutes ago

 [tests](#)

BLT-123: This is a good commit.

3 minutes ago

 [.drush-use](#)

BLT-123: This is a good commit.

3 minutes ago

 [.editorconfig](#)

BLT-123: This is a good commit.

3 minutes ago

 [.gitattributes](#)

BLT-123: This is a good commit.

3 minutes ago

 [.gitignore](#)

BLT-123: This is a good commit.

3 minutes ago

 [atlassian-ide-plugin.xml](#)

BLT-123: This is a good commit.

3 minutes ago

Install blt alias

and then you can use “**blt**”

```
$ composer blt-alias
```

Setup LAMP stack

Dev Desktop, DrupalVM, MAMP,
Native LAMP, etc.

```
$ blt vm
```

You have a virtual machine!

Let's do the other [local] stuff.

Do all the [local] things

- build dependencies
- install drupal
- import configuration
- set file permissions

```
$ blt setup
```


You have a fully functional Drupal site!

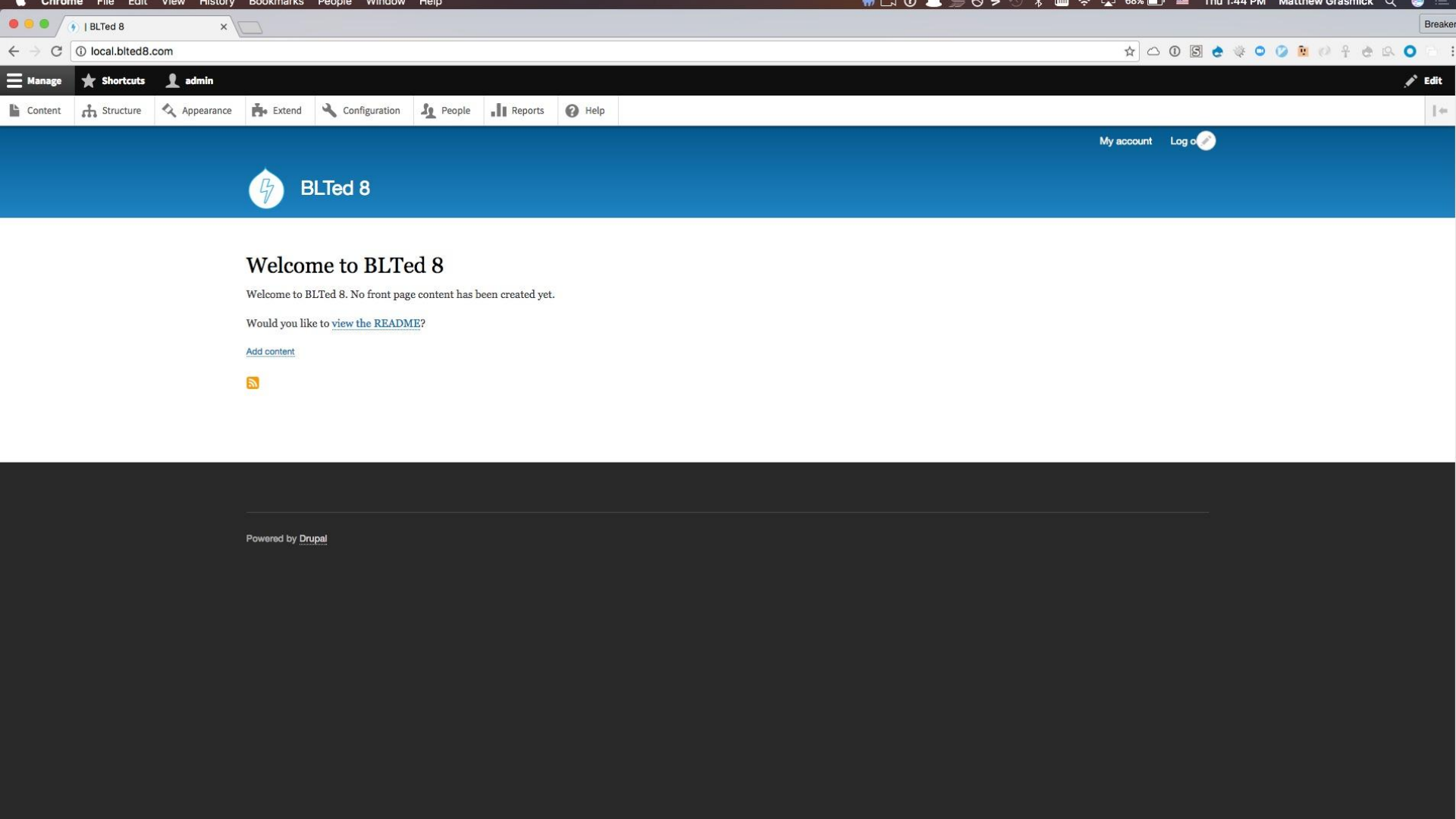
That's a lot of work
you didn't have to do.



Login to Drupal

because it's ready to go

```
$ cd docroot && drush uli
```



You've only run 4 commands.

Automated testing

Don't **freak** out.

We provide the tools and example tests

- Testing
 - Behat
 - PHPUnit
- Validation
 - PHPCS
 - Linting (PHP/Twig)

Validate code

phpcs, lint, composer

The right standards, on the right
extensions, in the right dirs

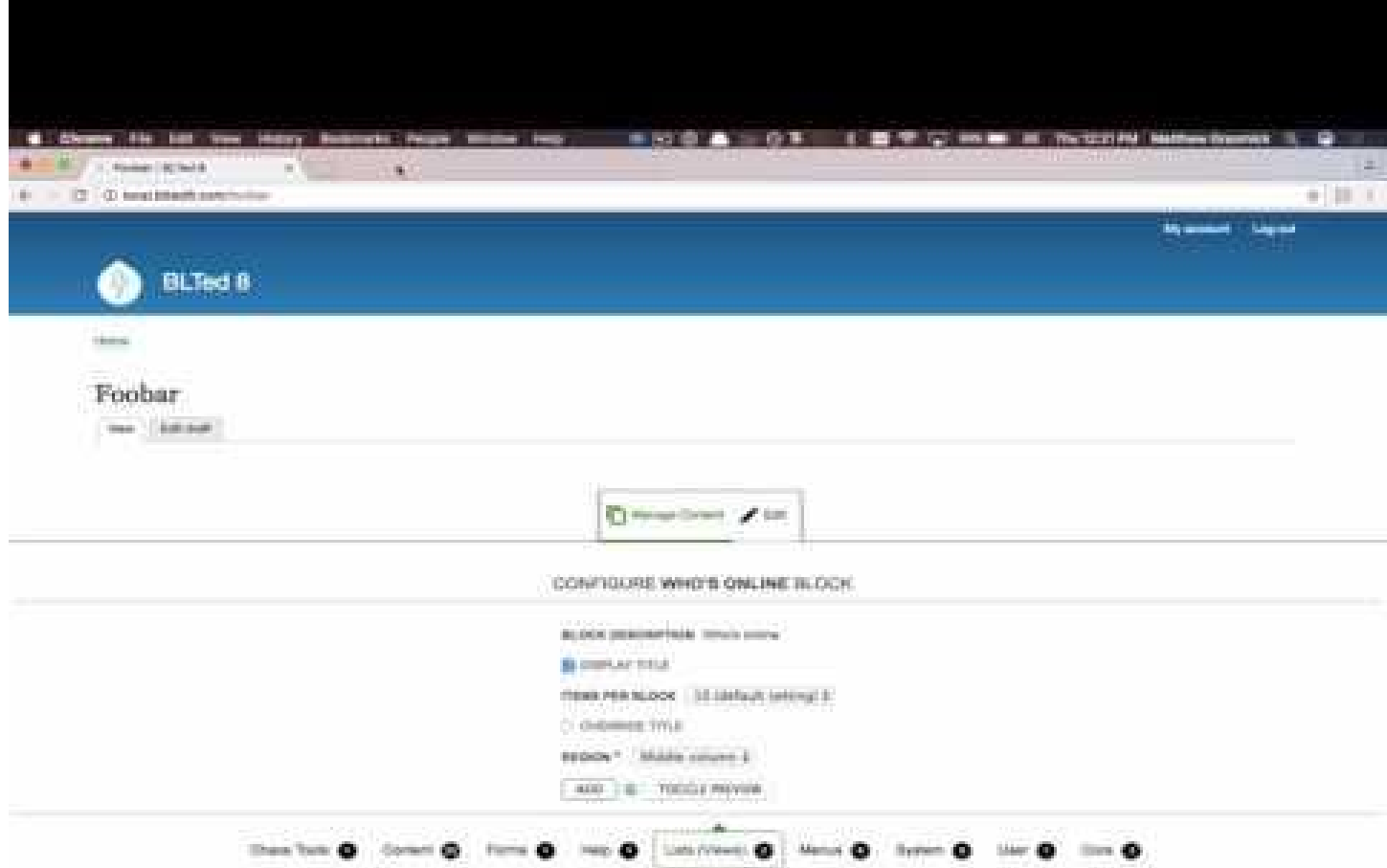
```
$ blt validate
```

Run them all!

Behat, PHPUnit all rolled up.

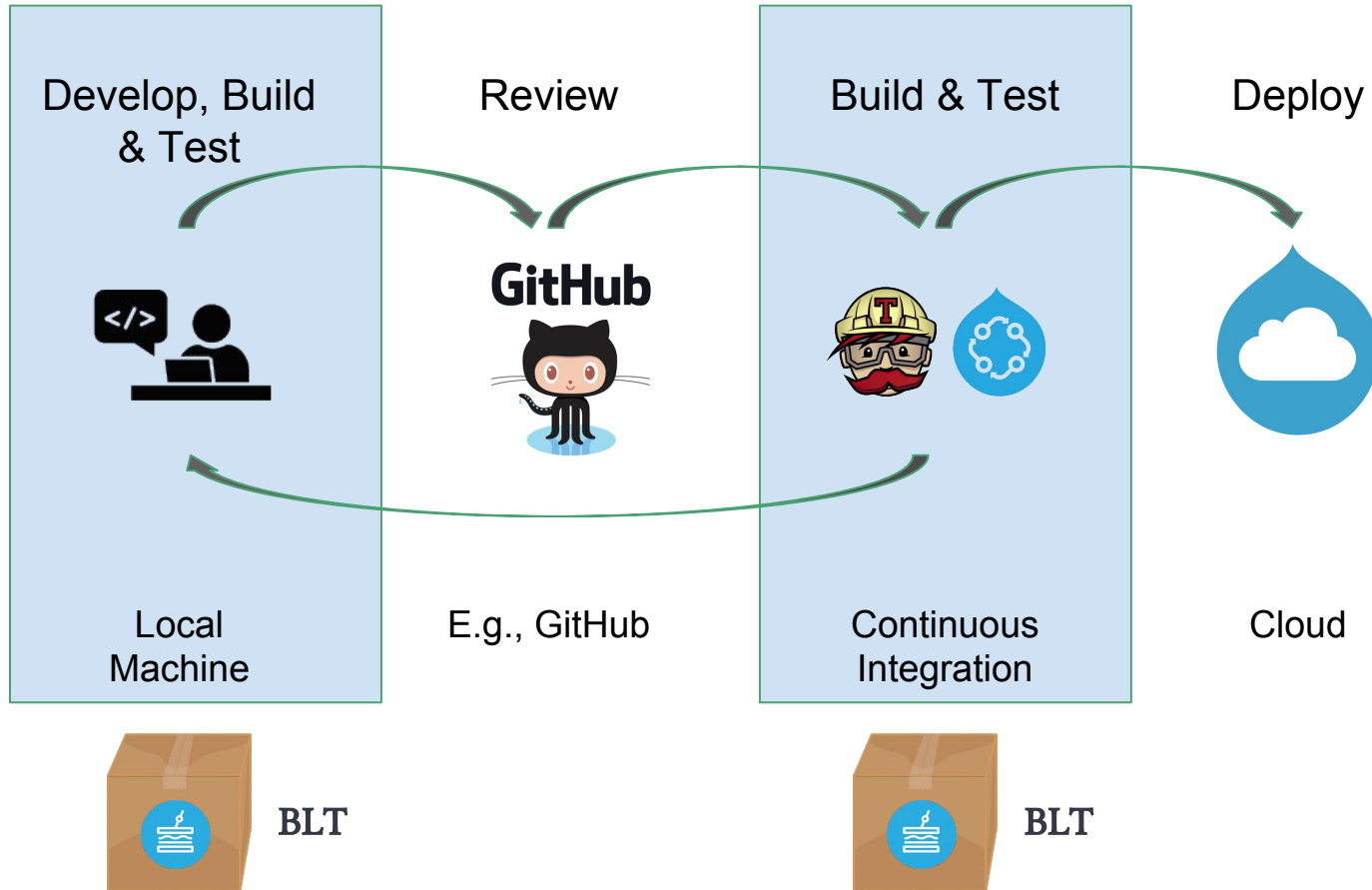
```
$ blt tests
```

Let's stop and appreciate this.



Let's talk about workflow

Not just local development



With this workflow...

Before merging, you:

- **Test** your changes automatically
- **Review** your changes on GitHub
- **Preview** your changes on a dedicated environment

Gives the confidence

To merge a deploy code

Adding Continuous Integration

Acquia Cloud CD and Travis CI out-of-the-box.

We provide CI config

You don't need to figure it out.

Acquia Cloud CD pipelines

acquia-pipelines.yml

```
$ blt ci:pipelines:init
```

You now have
acquia-pipelines.yml
pre-configured.

Travis CI

.travis.yml

```
$ blt ci:travis:init
```

You now have
.travis.yml
pre-configured.

Push

to the Cloud

```
$ git commit -a
```

```
$ git push origin
```

CI + BLT will build & test your site

Triggered by a **push** or **pull request**



This repository

Search

Pull requests Issues Gist



acquia / blt

Unwatch 21

★ Star 72

Fork 33

<> Code

Issues 25

Pull requests 11

Wiki

Pulse

Graphs

Settings

Filters

is:pr is:open

Labels

Milestones

New pull request

11 Open ✓ 124 Closed

Author

Labels

Milestones

Assignee

Sort

- ☐ **Added file path configuration.** in progress 2
#179 opened 3 days ago by danepowell
- ☐ **Minor Typo fix and updating default selenium port** 2
#178 opened 3 days ago by justinlevi
- ☐ **Add more placeholders to the behat example local configuration.** 4
#159 opened 5 days ago by damontgomery
- ☐ **Drupal composer source** in progress 0
#148 opened 7 days ago by grasmash
- ☐ **Make project description configurable.** 2
#140 opened 11 days ago by greylabel
- ☐ **Catch PHP messages with Behat.** in progress 3
#117 opened 13 days ago by danepowell
- ☐ **Adding test coverage for ACSF.** in progress 0
#111 opened 14 days ago by grasmash
- ☐ **Add missing cweagans/composer-patches requirement.** 8
#92 opened 20 days ago by geerlingguy
- ☐ **Refactor README.md** 0
#82 opened 25 days ago by theodorosproumis
- ☐ **Moves drush back [Do not merge until ACE 1.91 release]** 2
#34 opened on May 5 by kvleebrowning

Adding test coverage for ACSF. #111

Open grasmash wants to merge 1 commit into `acquia:8.x` from `grasmash:acsf-tests`

Conversation 0 Commits 1 Files changed 5

+7,852 -10



grasmash commented 14 days ago

Acquia member



No description provided.

Adding test coverage for ACSF.

b1fdf12

grasmash added the **in progress** label 14 days ago

Labels

in progress

Milestone

No milestone

Assignees

No one—assign yourself

1 participant

Notifications

Unsubscribe

You're receiving notifications because you authored the thread.

Lock conversation

Add more commits by pushing to the **acsf-tests** branch on **grasmash/blt**.



Some checks haven't completed yet

Hide all checks

1 pending check



continuous-integration/travis-ci/pr — The Travis CI build is in progress

Details



This branch has conflicts that must be resolved

Use the [command line](#) to resolve conflicts before continuing.



Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).



Search all repositories



My Repositories +

acquia/blt #234

Duration: 1 min 7 sec

Finished: -

✗ acquia/lightning #1395

Duration: 7 min 5 sec

Finished: about 3 hours ago

✓ acquia/df #431

Duration: 11 min 41 sec

Finished: 3 days ago

✓ acquia/statsgod #10

Duration: 2 min 45 sec

Finished: 3 days ago

✓ acquia/lightning-features #319

Duration: 7 min 30 sec

Finished: 4 days ago

✓ acquia/commons #26

Duration: 27 min 37 sec

Finished: 4 days ago

✓ acquia/moonshot #142

acquia / blt



build passing

Current

Branches

Build History

Pull Requests

Build #234

More options



Pull Request #111 Adding test coverage for ACSF.

#234 started



Commit c9a72af

#111: Adding test coverage for ACSF.



Elapsed time 1 min 7 sec



-

Matthew Grasmick authored and committed

Remove log

Raw log

1 Using worker: worker-linux-docker-1ad0655f.prod.travis-ci.org:travis-linux-8

2 Build system information

67 \$ export DEBIAN_FRONTEND=noninteractive

68 Adding ssh known hosts (BETA)

108 \$ git clone --depth=50 https://github.com/acquia/blt.git acquia/blt

136 This job is running on container-based infrastructure, which does not allow use of 'sudo',
137 setuid and setgid executables.

138 If you require sudo, add 'sudo: required' to your .travis.yml

139 See https://docs.travis-ci.com/user/workers/container-based-infrastructure/ for details.

140 \$ phpenv global 5.6 2>/dev/null

141 \$ phpenv global 5.6

142 \$ composer self-update

143 Updating to version 5a3d60c0cf0302a54b812f26fd23a0db67488e3e.

144 Downloading: connection... Downloading: 100%

145 Use composer self-update --rollback to return to version

146 1d8f05f1dd0e390f253f79ea86cd505178360019

147 Setting up build cache

148 \$ php --version

149 PHP 5.6.5 (cli) (built: Feb 12 2015 01:41:10)

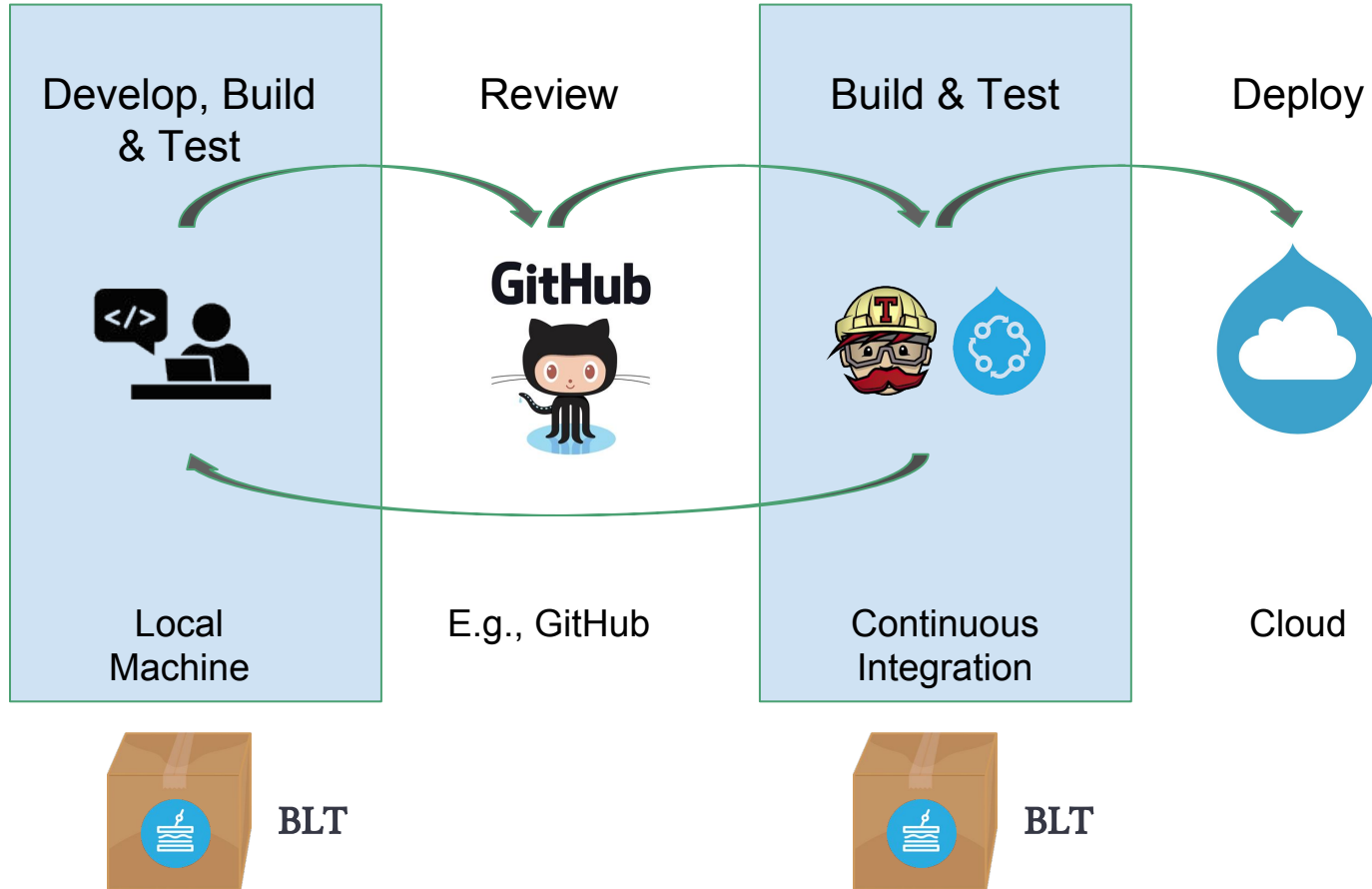
150 Copyright (c) 2007-2014 The PHP Group

Your tests run before merging

so you don't break things.

After merge, code deploys

all by itself.



Deploying to the Cloud

Deployment is special.

Only deploy artifacts!

Wait what? What's an artifact?

Things you need for
development process



Things you need for
production website

Development



Artifact



Development (SASS)

```
#navbar {  
  width: 80%;  
  border: solid red 1px;  
  float: left;  
  padding: 5px;  
  
  ul {  
    list-style-type: none;  
  }  
  li {  
    float: right;  
    a {  
      font-weight: bold;  
      color: red;  
    }  
  }  
}
```

Artifact (CSS)

```
#navbar {  
  width: 80%;  
  border: solid red 1px;  
  float: left;  
  padding: 5px;  
}  
  
#navbar ul {  
  list-style-type: none;  
}  
  
#navbar ul li {  
  float: right;  
}  
  
#navbar ul li a {  
  font-weight: bold;  
  color: red;  
}
```

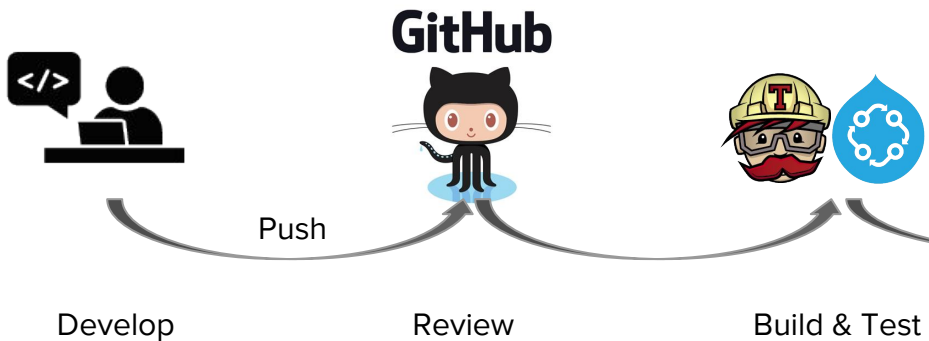
Development

```
.drush-use  
.git/  
.gitignore  
.idea/  
.travis.yml  
.vagrant/  
README.md  
Vagrantfile  
box/  
build/  
composer.json  
composer.lock  
config/  
docroot/  
drush/  
drush.wrapper*  
example.project.local.yml  
factory-hooks/  
hooks/  
patches/  
project.local.yml  
project.yml  
readme/  
reports/  
scripts/  
tests/  
vendor/
```

Artifact

```
.gitignore  
composer.json  
composer.lock  
config/  
docroot/  
drush/  
factory-hooks/  
hooks/  
scripts/  
vendor/
```

Working Repo



Artifact

Dev	Stage	Prod
<div>Code develop</div> <div>Files</div> <div>DB</div>	<div>Code tags/2016-06-17</div> <div>Files</div> <div>DB</div>	<div>Code tags/2016-06-17</div> <div>Files</div> <div>DB</div>

Build Artifact & Push

Why are we doing this?

Lots of good reasons.

Improves process, improves product.

- **Maintainability**

- Prevents hacking contrib & core (not committed).
- Deployments are scripted and repeatable (update hooks are required).

- **Performance**

- Removes unnecessary tools and libraries.

- **Security**

- Artifacts are sanitized.

Show me.

Manual deployment is good for debugging.

deploying manually

from your local machine

```
$ blt deploy
```

What just happened?

- A deployment artifact was created at your/project/path/**deploy**
 - Files were copied.
 - Front end assets were compiled.
 - Prod dependencies were built.
 - Directory was sanitized.
 - **Everything was committed.**
- The deployment artifact was:
 - Committed to the “<original-branch>-build” branch.
 - **Pushed to** `$(project.git.remotes)`

Recap

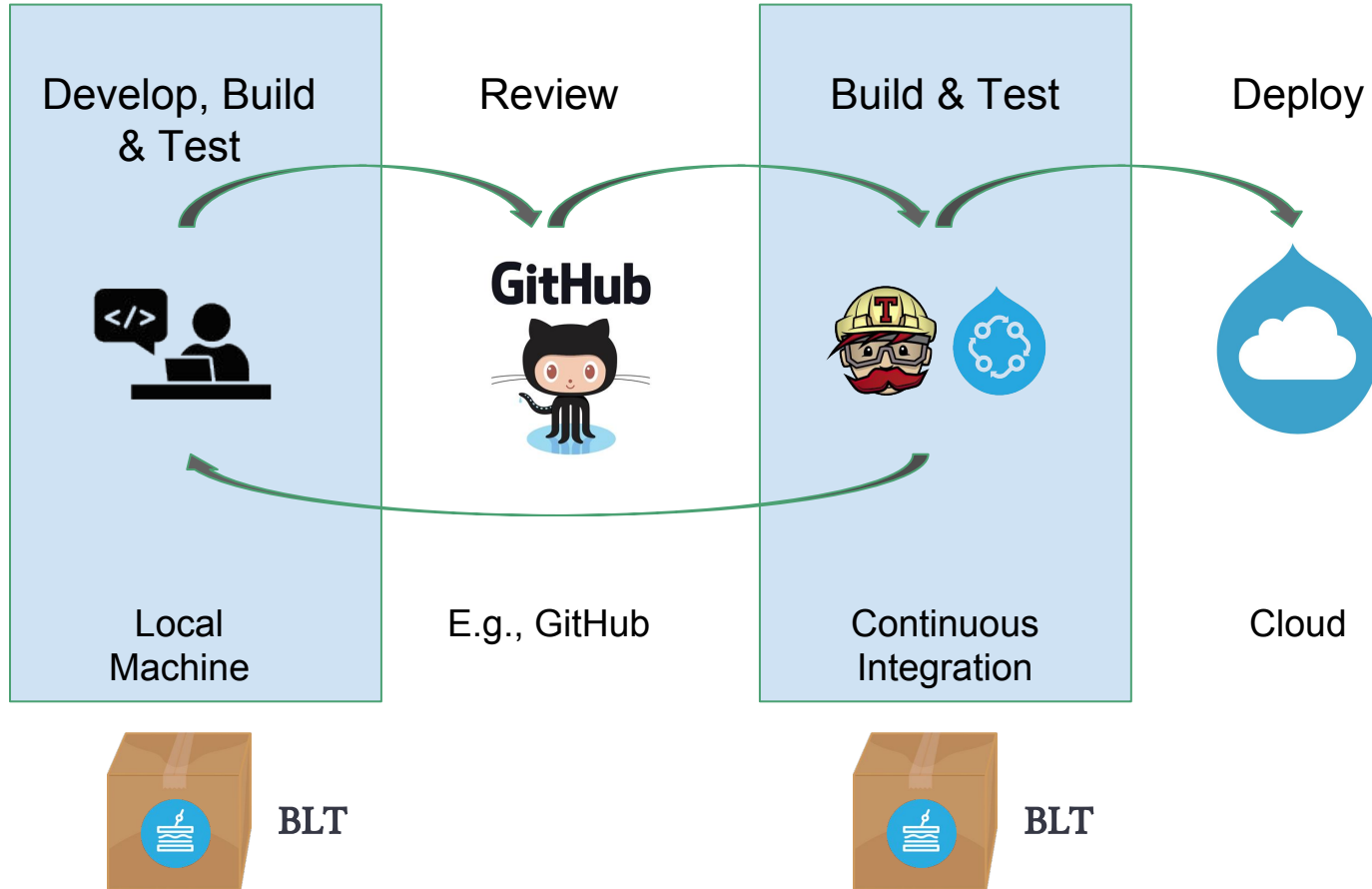
BLT is a box with:

- Standardized template for Drupal 8 sites
- Set of tools for building, testing, deploying
- Commands for automating usage of those tools

Make complex things simple

- **Create** an entire project
- **Boot** (& create) a VM
- **Validate** all of your code
- **Test** your application
- **Deploy** to cloud

... in one command



Thanks!

Questions?

Reminder: Sprints are tomorrow!

github.com/acquia/blt

[@grasmash](#)

bit.ly/2oa9j1w
