DurableDrupal Lean

Tool up for solid web apps now and maintain them for years with a powerful Drupal distro

Victor Kane

DurableDrupal Lean

Tool up for solid web apps now and maintain them for years with a powerful Drupal distro

Victor Kane

Contents

Preface

This is preface stuff

Introduction

Tool up for solid web apps now and maintain them for years with a powerful Drupal distro

No-one wants websites any more. This book is for web app development teams, called upon by their clients to move their business model to the internet Because that's what's going on (even if, like most people, they still talk about "websites").

This book will empower you and your team, no matter what your level of expertise, to simplify the building of solid solutions for your clients' real needs, and to facilitate the maintenance these solutions will be crying out for from the get go.

In the process, you will actually be evolving for yourselves a lean process factory to output durable solutions that can be maintained for years to come.

Because the real costs are all about maintenance and constant change:

- First in the minimum of 40% in changes that will be requested in the weeks or months of development.
- Secondly, in the feature and usability requests the solution will have to assimilate in order to maintain its value over time.
- Thirdly in the security threats that will have to be dealt with in as fully an automated way as possible.
- Fourthly in the architectural and deployment changes that will be required over the years, in the struggle to scale with growth and cut costs with new technological breakthroughs.

Because it's not about the "single click site generation" and easy scaffolding any decent framework or "hosting" solution offers. We'll soon see how to fire up a framework in seconds, in just a few clicks, but the devil is in how we empower ourselves with a process that is going to facilitate, stabilize and simplify our work in the face of constant change we need to flow with.

We're here for the long haul.

And we're using Drupal[1] now.

So here's what we'll be doing:

[1] https://www.drupal.org/

- Chapter 1: Overcoming Choice Panic - What frameworks should we use?

  We need to free ourselves from choice panic which can paralyze us or disrupt our work at any time if the major project stakeholders aren't all on board with the major framework and other choices that will be made during the appropriate iterations. We need to have this discussion sooner rather than later, and realize that as time goes on we'll always be migrating. Realizing that will allow us to prepare gradually and ease the pain. Hey, it's part of the job!
  - This is what we need and we need it now
    We need to adopt a process which bases itself brutally on what we need and what our clients need, and which abandons vested vendor lock-in of all kinds. We can't be married to frameworks: it's one big open-source community and its one big set of solutions we must achieve
  - Building from scratch or not? Frameworks or not?
  - Proprietary or not? Open-source or not?
  - WordPress or not?
  - Drupal or not?
  - Drupal 8 or not?
  - Backdrop or not?
  - Vanilla Drupal 7 or not? Distro or not?
  - Presenting DurableDrupalDistro: fork it, tailor it, maintain it, branch and release it over all your projects!
  - Preparing to migrate gradually all along, so we'll be ready when the time comes
- Chapter 2: Setting up just enough shop to deliver our first web app
  - Every team member has their own dev environment
    - On Pantheon
    - On Amazon
  - What the client needs
    - The tangle of User Stories and Design
  - Fork the distro and make it our own
  - User stories and doing it with agile
  - Delivery as client empowerment
  - Post mortem: We did it! But...
    - Agile is the new waterfall (pixel perfect design -> "implementation")
    - Team couldn't really work in parallel
    - Client took delivery of a monolithic blob he wasn't sure how to use
      - We had to do special training (not to mention train the trainers)
      - Client began a slew of change requests that could have come earlier if they had been familiar with the system
- Chapter 3: Agile is the new waterfall - we need Lean!
  - New project!
  - Value hypotheses
  - Design studio for nerds... and for the client too
  - Getting the team working in parallel

- Chapter 4: Functional prototypes
- Chapter 5: Grabbing the legacy content
- Chapter 6: Candidate architecture
  - Choices panic revisited
  - We're evolving on top of best of breed
- Chapter 7: Build the recursive MVP of MVP's
  - Every user story is an MVP
  - No project is ever anything more than an MVP
- Chapter 8: Go team test team go go go

  The advantage of using frameworks isn't just that they're tried and tested, but that they empower testing and TTD
- Chapter 9: Build, deploy, maintain
  - Rinse and repeat over "development"

But more than a book this is a community of teams. Dive in!

Chapter 1 - Overcoming Choice Panic - What frameworks should we use?

This is what we need and we need it now

We're a new web development team and we have our first job together. In the best agile tradition it's a "light" one, so that we can get to know each other and build some synergy to prepare us for a bigger project we'll be starting in a month or so (an online magazine). A neighborhood non-profit runs programs for youth in the community. They've had a webmaster who managed the content on their site for them, but he seems to have disappeared and they've lost access to their old stuff. Actually they've been wanting to re-do their site for some time, so they have some ideas, but they're stuck. They want to be able to manage their content without depending on anyone else for that.

So what should we use? After discarding some options let's say the list boils down to the following:

- Buy a proprietary solution
- Use Open Source:
  - Write our own from scratch
  - WordPress[2]
  - Drupal[3]

The trouble is, there are cases in which any one of these choices could be the right one. But since we're starting out we want something we could learn to use together and then keep on using as we grow.

[2]https://wordpress.org/
[3]https://drupal.org/

Proprietary vs Open Source

Existing frameworks vs rolling our own

Drupal vs WordPress

So which Drupal should we use?

Drupal 8

Backdrop

Drupal 7

Won't everything we build in Drupal 7 be obsolete in a few months?

It's all in the maintenance

- It's not just about building, we have to be experts in maintenance
- …
- Discussion about why we need a distro

Let's not just use Drupal 7, let's use a re-usable distro that we can evolve over time and even migrate… something that will last us several years

Create a distro with the hotness of Drupal 8 but without the waiting or the learning curve

Everything in code with install profile and configuration management ### Migration of content ### Easy to set up development environments for all team members using continuous build ### Agile and lean process

Introducing DurableDrupalDistro with Lean process

….. in this chapter we …. In Chapter 2 - DurableDrupalDistro, we'll …

Chapter 2 - DurableDrupalDistro

El Viejo certainly had our attention.

"We'll first show you how to set up a workspace[4], getting a folder somewhere you can configure as a Drupal doc root, that is, the main Drupal directory that can be picked up by a LAMP stack and made available to a browser as a URL.

This can be done on a dedicated server, on a VPS, on a virtual machine or directly on your laptop, even on a good shared hosting, but you've got to have ssh access to the command line.

We all need to be able to do this no matter what our role on the team, so we'll show you a couple of examples.

Then we'll see how to grab our very own DurableDrupalDistro and fire it up[5].

Finally, we'll check out what we have going for us[6] under the hood right off-the-shelf, we'll take our first steps[7], and then I'll give an example of updating and maintaining the distro itself[8]".

Getting a workspace in a LAMP stack

El Viejo shows here how to set up a workspace from scratch, in his usual opinionated fashion.

- on a VPS like Linode or Digital Ocean,
- on personal laptop/workstation alternatives
  - Water Works stack
  - Bitnami
    - Drupal App
    - Drupal Server
  - Kalabox
- Roll your own
- on a shared hosting account with command line access and drush

VPS

TODO linode

TODO digital ocean

[4]a_workspace_in_a_lamp_stack.html
[5]obtaining_and_installing_the_durabledrupaldistro.html
[6]whats_included_in_durabledrupaldistro.html
[7]durabledrupaldistro_first_steps.html
[8]updating_durabledrupaldistro_itself.html

Water Works stack

Well, Water Works never got around to offering one, or else they did and then started charging for it, so nobody used it and it died. But Acquia has one! Very convenient for running Drupal apps on your laptop, but it isn't a virtual machine, it's an app only, so that means that while you can run it and everything, it doesn't emulate the target live environment (a GOOD idea) the app will be running on when it is launched. Get it here[9].

Bitnami

Bitnami[10] swings both ways, with the app approach and the VirtualBox VM alternative (recommended but a lot of work[11] and then more work[12]).

Pantheon

Single click …..

And you can have the best of both worlds, with pantheon in the cloud and on your desk… Kalabox!

Kalabox

Ah, the amigos at Kalamuna[13] are a breath of fresh air. They've not only come up with Kalabox[14] but are now developing Kalabox 2[15], the NodeWebkit[16] and Docker[17] wonder, as a completely open source project after a successful Kickstarter campaign[18].

I've tried it out quite a bit. See how I installed Kalabox 1[19] and used it as a Pantheon workflow solution. Integration with Pantheon hosting[20] and their one-click Drupal Distro site builder is a hallmark of Kalabox 1. Very useful considering its revolutionary architecture[21] and standardizing workflow[22] (at least as far as Drupal hosting is concerned). Alternative hosting support plug-ins are expected with Kalabox 2.

[9] http://www.acquia.com/downloads
[10] https://bitnami.com/stack/drupal
[11] http://awebfactory.com/node/524
[12] http://awebfactory.com/node/525
[13] http://www.kalamuna.com/
[14] http://www.kalamuna.com/products/kalabox
[15] https://github.com/kalabox
[16] https://github.com/rogerwang/node-webkit
[17] https://www.docker.com/
[18] https://www.kickstarter.com/projects/kalabox/kalabox-advanced-web-tools-for-the-people
[19] http://awebfactory.com/node/522
[20] https://www.getpantheon.com/
[21] https://www.getpantheon.com/how-it-works
[22] https://www.getpantheon.com/how-it-works

Ansible

(study up on Ansible for Dev-Ops book first! buy the bundle!) * Ansible on a Mac to run a server * Ansible on a Mac to run a local vagrant/vbox instance

Roll your own

sthg about virtual box & vagrant, with puppet, chef, ansible…

Shared hosting account

We shouldn't even be talking about this, but recent improvements in shared hosting accounts have made something like a reseller account somewhat (only just) viable, at least for hosting small to middling projects.

First thing to do is get Drush on there, once you've gotten ssh command-line access, if you can't get that switch hosting now (and if you're even reading this section, second thing is to create a subdomain for each site: don't go running Drupal from a sub-directory).

Obtaining and installing the DurableDrupalDistro

What's included in DurableDrupalDistro

DurableDrupalDistro: First Steps

So we're in oue workspace, let's say on a VPS.

What we want to do

- Create virtual host
- Create database
- Clone and set up distro
- Install with drush on the command line
- Check it out

Create virtual host

Create database

Clone and set up distro #### Install with drush on the command line #### Check it out

Updating DurableDrupalDistro itself

```
$ drush pm-refresh
Refreshing update status information ...
Done.
$ drush pm-update
 Name                 Installed  Proposed  Message
                      Version    version
 Drupal               7.31       7.32      SECURITY UPDATE available
 Calendar (calendar)  7.x-3.4    7.x-3.5   Update available
 CKEditor (ckeditor)  7.x-1.15   7.x-1.16  SECURITY UPDATE available
 Profiler Builder     7.x-1.1    7.x-1.2   Update available
 (profiler_builder)



Update information last refreshed: Tue, 10/21/2014 - 13:20
NOTE: A security update for the Drupal core is available.
Drupal core will be updated after all of the non-core projects are updated.

Security and code updates will be made to the following projects: Calendar [cale\
ndar-7.x-3.5], CKEditor - WYSIWYG HTML editor [ckeditor-7.x-1.16], Profiler Buil\
der [profiler_builder-7.x-1.2]

Note: A backup of your project will be stored to backups directory if it is not \
managed by a supported version control system.
Note: If you have made any modifications to any file that belongs to one of thes\
e projects, you will have to migrate those modifications after updating.
Do you really want to continue with the update process? (y/n):
Project calendar was updated successfully. Installed version is now 7.x-3.5.
Backups were saved into the directory                                     [ok]
/home/drupallean/drush-backups/drupal_lean/20141021132817/modules/calendar.
Project ckeditor was updated successfully. Installed version is now 7.x-1.16.
Backups were saved into the directory                                     [ok]
/home/drupallean/drush-backups/drupal_lean/20141021132817/modules/ckeditor.
Project profiler_builder was updated successfully. Installed version is now 7.x-\
1.2.
Backups were saved into the directory                                     [ok]
/home/drupallean/drush-backups/drupal_lean/20141021132817/modules/profiler_build\
er.

Code updates will be made to drupal core.
WARNING:  Updating core will discard any modifications made to Drupal core files\
, most noteworthy among these are .htaccess and robots.txt.  If you have made an\
y modifications to these files, please back them up before updating so that you \
can re-create your modifications in the updated version of the file.
Note: Updating core can potentially break your site. It is NOT recommended to up\
date production sites without prior testing.

Do you really want to continue? (y/n):
Project drupal was updated successfully. Installed version is now 7.32.
Backups were saved into the directory                                  [ok]
/home/drupallean/drush-backups/drupal_lean/20141021132817/drupal.
No database updates required                                           [success]
'all' cache was cleared.                                               [success]
Finished performing updates.                                           [ok]
$ git status
$ git add .
$ git commit -am "Maintenance update to drupal core 7.32 plus updated contrib mo\
dules"
$ git tag
$ git tag -a "October2014" -m "Maintenance core and contrib update"
$ git push --tags
```

Now we need to prepare the site (see Quick install for developers (command line)[23]

```
cp sites/default/default.settings.php sites/default/settings.php
```

Give the web server write privileges (666 or u=rw,g=rw,o=rw) to the configuration file.

```
chmod a+w sites/default/settings.php
```

Give the web server write privileges to the sites/default directory.

```
chmod a+w sites/default
```

so we can re-install from scratch to test. Check out incredible options available with drush help:

[23]https://www.drupal.org/documentation/install/developers

```
$ drush help site-install
Install Drupal along with modules/themes/configuration using the specified
install profile.

Examples:
 drush site-install expert --locale=uk     (Re)install using the expert install
                                           profile. Set default language to
                                           Ukranian.
 drush site-install                        Install using the specified DB
 --db-url=mysql://root:pass@localhost:por  params.
 t/dbname
 drush site-install                        Install using SQLite (D7+ only).
 --db-url=sqlite://sites/example.com/file
 s/.ht.sqlite
 drush site-install --account-name=joe     Re-install with specified uid1
 --account-pass=mom                        credentials.
 drush site-install standard               Pass additional arguments to the
 install_configure_form.site_default_coun  profile (D7 example shown here - for
 try=FR                                    D6, omit the form id).
 my_profile_form.my_settings.key=value
 drush site-install                        Disable email notification during
 install_configure_form.update_status_mod  install and later. If your server
 ule='array(FALSE,FALSE)'                  has no smtp, this gets rid of an
                                           error during install.


Arguments:
 profile                                   the install profile you wish to run.
                                           defaults to 'default' in D6,
                                           'standard' in D7+
 key=value...                              any additional settings you wish to
                                           pass to the profile. Fully supported
                                           on D7+, partially supported on D6
                                           (single step configure forms only).
                                           The key is in the form [form
                                           name].[parameter name] on D7 or just
                                           [parameter name] on D6.


Options:
 --account-mail                            uid1 email. Defaults to
                                           admin@example.com
 --account-name                            uid1 name. Defaults to admin
 --account-pass                            uid1 pass. Defaults to a randomly
```

```
                                                generated password. If desired, set
                                                a fixed password in drushrc.php.
 --clean-url                                    Defaults to 1
 --db-prefix                                    An optional table prefix to use for
                                                initial install.  Can be a key-value
                                                array of tables/prefixes in a
                                                drushrc file (not the command line).
 --db-su=<root>                                 Account to use when creating a new
                                                database. Must have Grant permission
                                                (mysql only). Optional.
 --db-su-pw=<pass>                              Password for the "db-su" account.
                                                Optional.
 --db-url=<mysql://root:pass@host/db>           A Drupal 6 style database URL. Only
                                                required for initial install - not
                                                re-install.
 --locale=<en-GB>                               A short language code. Sets the
                                                default site language. Language
                                                files must already be present. You
                                                may use download command to get
                                                them.
 --site-mail                                    From: for system mailings. Defaults
                                                to admin@example.com
 --site-name                                    Defaults to Site-Install
 --sites-subdir=<directory_name>                Name of directory under 'sites'
                                                which should be created. Only needed
                                                when the subdirectory does not
                                                already exist. Defaults to 'default'

Aliases: si
```

So let's install in Spanish, specifying db credentials, uid1 user name and password, uid1 user email, site email and site name

```
site-install drupallean --locale=es --db-url=mysql://drupal_lean:lean@localhost/\
drupal_lean --account-name=admin --account-pass=omg --account-mail=victorkane@gm\
ail.com --site-mail=drupallean@awebfactory.com.ar --site-name=DrupalLean

You are about to DROP all tables in your 'drupal_lean' database. Do you want to \
continue? (y/n): y
No tables to drop.                                                    [ok]
Starting Drupal installation. This takes a few seconds ...            [ok]
WD php: Warning: Invalid argument supplied for foreach() in           [warning]
```

```
_features_restore() (line 966 of
/home/drupallean/drupal-lean/sites/all/modules/contrib/features/features.module).
Installation complete.  User name: admin  User password: omg     [ok]
```

Chapter 3 - Value Hypotheses for the project

Chapter 4 - Functional prototypes

Chapter 5 - Grabbing the legacy content

Chapter 6 - Candidate architecture

Chapter 7 - Build the recursive MVP of MVPs

Chapter 8 - Go team test team go go go

> If I hear “Get out of the building” one more time I think I’ll scream

It’s not just about code that runs right, even though it must.

It’s about value! People’s needs!

Chapter 9 - Build, deploy, rinse, repeat

> If I hear “Rinse and repeat” one more time I think I’ll scream

“Whatever happened to getting it right the first time?” muttered Jason, sick and tired of stuff he thought was already done being “postponed” to “the next iteration”.

Appendix 1 - how to

how to