PROFESSIONAL WORDPRESS DEPLOYMENT

By @ramisayar

HI WORDCAMP!

@ramisayar
github.com/sayar

BACKGROUND

Infrastructure and backend dev in SF + MTL.
I do python, php, and node.js.

WHAT DO I DO NOW?

I write code for fun and clients @SayarLabs in MTL.

Mostly large scale digital development.

WHAT'S THIS ABOUT?

Professional Software, Development and deployment.
What does it mean? Why is it important?

RELIABILITY

Professional Software is technology that one can rely on to be up and working every hour of the day, right when you need it most. No excuses, no BS.

MAINTENANCE
SCALE
SUPPORT

Professional Software is technology that you run, maintain, scale and support indefinitely until the end of it's usefulness.

SCM
VERSIONED
MANAGED RELEASES

Professional Software Development is development that takes care of source code management, doesn't test changes in production, let's you branch and merge.

STRONG ENVIRONMENT GUARANTEES

Professional Software Development is when you can simulate environments, when you have strong guarantees on your target platform, when sh*t just works as expected, no surprises.

SOLID REPEATABLE STACKS

Professional Software Development is when you can set up your stack repeatably, when the stack is versionned, when `apt-get` doesn't break random stuff.

AUTOMATED DEPLOYMENT

Professional Software Development is when deployment is automated and doesn't take 8 hours with every one on the edge of their seats.

LOGGING DONE RIGHT

Professional Software is technology that is auditable, logs correctly, remembers to rotate logs and persists those logs for analysis.

MONITORABLE
SECURE
DEVELOPER FRIENDLY

PROFESSIONAL SOFTWARE DEVELOPMENT IS WHEN DEVOPS IS PART OF THE CULTURE.

SO LET'S TALK

WordPress

DO WE HAVE A DEVOPS PROBLEM?

WORDPRESS IS AWESOME!

BUT LARGE CLIENT INSTALLS, HMM...

WORLD-CLASS FRONT ENDS NEED WORLD-CLASS BACK ENDS.

PROBLEMS?

- Lack of an established set of practices and techniques for professionalizing WordPress development and deployment.
- Few resources for beginners to deploy WordPress like Pros.

WE CAN FIX THIS EASY!

DEVOPS APPLIES TO ALL THINGS AND EVERYTHING.

LET'S SHARE!

PROWP.ORG

WHAT'S DOES IT LOOK LIKE NOW?

Famous 5 Minute Install -> YES, this is awesome but...
Hosting Panels -> One button install... ugh.

STOP THIS MADNESS!

LET'S GET STARTED!

SOURCE CODE MANAGEMENT

- Choose an SCM, such as git or svn.

```
$ git clone -b 3.5-branch git@github.com:WordPress/WordPress.git
```

- Git Submodules are your new best friend.

```
cd Wordpress
git submodule add git@github.com:facebook/wordpress.git wp-content/plugins/facebo
```

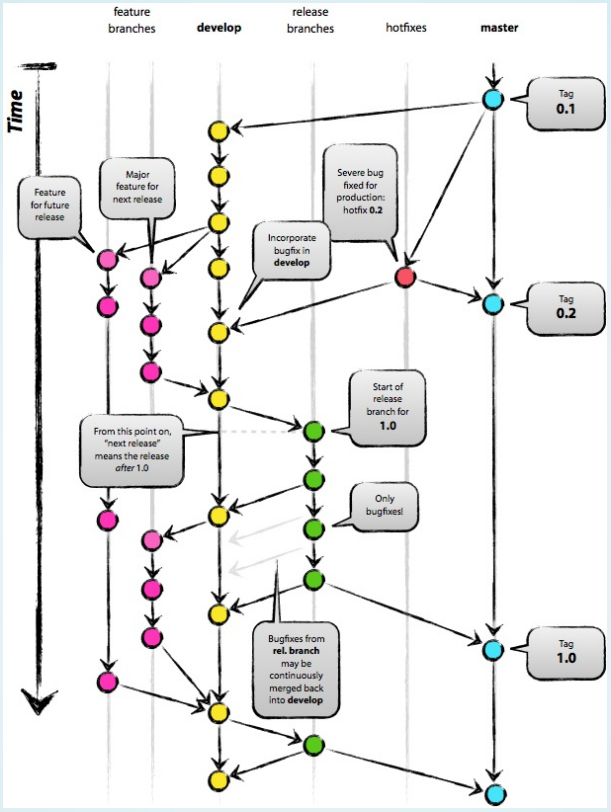
- Do the same for your theme.

SOURCE CODE MANAGEMENT

- WAIT! Fork the WordPress repository first! Run your own master branch (on GitHub or own server).
- Run on master or a release branch.
- Set original WordPress repository as upstream and sync up per release.

```
git clone git@github.com:sayar/WordPress.git -b 3.5-branch
cd WordPress
git checkout -b dev
git remote add upstream git@github.com:WordPress/WordPress.git
# Update WP when a new release comes out
git fetch upstream
git merge upstream/3.5-branch
git push origin dev
```

SOURCE CODE MANAGEMENT



Vincent Driessen - http://nvie.com/posts/a-successful-git-branching-model/

ENVIRONMENT SIMULATION

- Simulate your production environment locally using Vagrant.
  - http://downloads.vagrantup.com/
  - sudo apt-get install vagrant
- Vagrantpress is an excellent start.
  - Ubuntu 12.04 (64bit) VM
  - Apache2
  - PHP5
  - MySQL
  - WordPress (GIT!)
  - Puppet (Coming back to this)

ENVIRONMENT SIMULATION

- How can we use Vagrantpress?

```
git clone https://github.com/chad-thompson/vagrantpress
cd vagrantpress
vagrant up
```

- This will download (takes a while... please do this at home) and start the box.
- Visit http://localhost:8080/wordpress/

ENVIRONMENT SIMULATION

- Setting up your git repository in the box (assuming public github, if not set up read-only deploy keys on your box).

```
vagrant ssh
cd /vagrant/wordpress/
git remote rm origin
git remote add origin https://github.com/sayar/WordPress.git
git fetch origin
git checkout -t origin/dev
```

- To update your environment:

```
vagrant ssh
cd /vagrant/wordpress/
git pull origin dev
```

ENVIRONMENT DISTRIBUTION

BRAND NEW TOOL - LAUNCHED YESTERDAY!

PACKER!

FROM THE CREATOR OF VAGRANT.
PACKER.IO

PACKER

Packer is an open source tool for creating identical machine images for multiple platforms from a single source configuration.

Packer does not replace configuration management like Chef or Puppet. In fact, when building images, Packer is able to use tools like Chef or Puppet to install software onto the image.

PACKER

- Integrating Packer into our Environment Simulation Workflow to achieve parity between production and development.
- Use Packer to Create Images for AWS (or others).
- Packer builds the machine image.
  - Use Chef or Puppet to configure the image. (More later)
  - Keep your image around for all your future WordPress project needs.
- Post-process Packer Machine Image into a Vagrant Box!
- Run Vagrant Box locally... push packer image to cloud.

AUTOMATED ENVIRONMENT SETUP

- Automated and repeatable environment setup
- Several options available:
  - Chef
  - Puppet (supported by vagrantpress)
- If Chef or Puppet are complicated for your use cases:
  - etckeeper is a collection of tools which let you store /etc in git (or another DVCS)
  - etckeeper will let you store your apache, php, mysql configs and automate your environment setup and ensure its consistency.

CHEF

- Store Configuration in Recipes
- Each node has chef-client.
- Hosted Chef server pushes configuration to nodes.
- There exists a repository of reusable cookbooks.

- Chef is awesome if you more than 3 servers...

AUTOMATED TESTING

- A small distraction before we discuss testing.
- wp-cli should be installed by default on your machine.

> WP-CLI is a set of command-line tools for managing WordPress installations. You can update plugins, set up multisite installs and much more, without using a web browser.

INSTALL IT!

AUTOMATED TESTING

- With wp-cli installed, you can generate unit-test scaffolding for plugins.
- Place scaffolded code in its own git submodule. ;)

```
wp core init-tests
wp scaffold plugin-tests
```

- If you're hosted publicly on GitHub (or otherwise), you can use Travis CI or Jenkins to automate testing on every git push.

APPLICATION MONITORING

- In general, two strategies can be used, either a hosted solution by a third-party or running your own metrics infrastructure.
- For a hosted solution, you can use a fairly popular service called New Relic.
- Running your own infrastructure, you can use Nagios.

DATABASE MONITORING

- Whole different ballgame. Endless MySQL monitoring applications available for use.

MIGRATION MANAGEMENT - STAGING

- Host migration is a real problem on highly used sites.
- Testing staging releases with real database access can be problematic.

CODE FREEZE TO THE RESCUE!

WORDPRESS SECURITY...

CACHING...

MEMCACHED + BATCACHE FTW!

BATCACHE

Batcache uses Memcached to store and serve rendered pages. It can also optionally cache redirects. It's not as fast as Donncha's WP-Super-Cache but it can be used where file-based caching is not practical or not desired. For instance, any site that is run on more than one server should use Batcache because it allows all servers to use the same storage.

BATCACHE

- Benefits:
  - Can rely on cloud memcached services (e.g. AWS)
  - Removes run-time generation of cached files.

- Disadvantages:
  - Not as fast as Super-Cache but definitely cleaner.

EXTRAS - AUTOMATTIC DEVELOPER PLUGIN

- Debug Bar
- Debug Bar Console
- Debug Bar Cron
- Debug Bar Extender
- Rewrite Rules Inspector
- Log Deprecated Notices
- Log Viewer - This plugin provides an easy way to view *.log files directly in the admin panel.
- Monster Widget
- User Switching
- Beta Tester

MIGRATING TO THE CLOUD

Caching -> AWS ElastiCache
Servers -> AWS EC2
Load Balancers -> AWS ELB
Databases -> AWS RDS

Monitor, deploy and do a lot more repeatably and in an automated fashion while scaling.

WHAT DID WE DO?

PROFESSIONAL WORDPRESS DEVELOPMENT AND DEPLOYMENT!

Now go scale!

THANK YOU!

ANY QUESTIONS?

@ramisayar - github.com/sayar

http://slid.es/sayar/prowp