



DRUPALCAMP
SPAIN 2015
JEREZ DE LA FRONTERA

Drupal Extreme Scaling



About Me

Academics

- Computer Science Engineer - UCA
- Masters Degree on Software Engineering - US
- Masters Degree on IT Security - US

Experience

- 5+ years as web developer, 3 of them using Drupal
- Currently: Freelance DevOps
- IT Security, GNU/Linux, Python, Node.JS . . .

Interests

- Rock'n'Roll (electric guitar) and videogames
- Books, movies, beer, whiskey, roleplaying . . .

1 Introduction

2 The Project

3 Problems and Solutions

4 Demo

5 Conclusions

Contents

1 Introduction

2 The Project

3 Problems and Solutions

4 Demo

5 Conclusions

How does Drupal scale?

- What is scaling?
- Related to performance, but it's not the same
- Horizontal scaling
- Vertical scaling
- Drupal is highly cache-dependent



Drupal performance, scalability & availability

- Memcache
- Apc / Opcache
- Varnish
- Redundancy
- Choose your cache improvement module



Cloud computing and containers

- Elastic computing → resources grow on demand
- In theory, we achieve full scalability and availability. **False!**
- The hidden enemy: budget grows till infinity and beyond
- Containers (Docker) → Only for local environments?



Contents

1 Introduction

2 The Project

3 Problems and Solutions

4 Demo

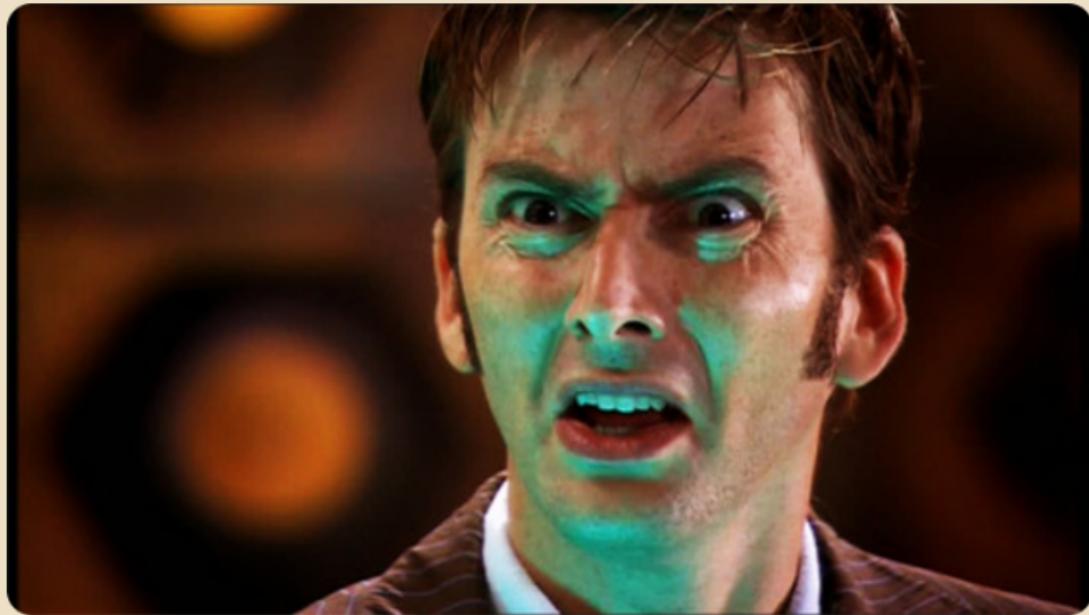
5 Conclusions

The Project

Some base requirements

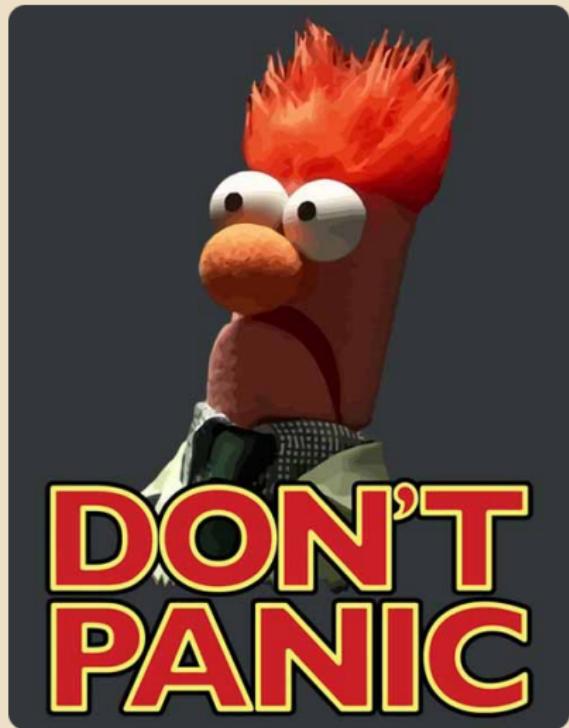
- Built in Drupal 7, as a multisite install
- Currently, +30k sites. It needs to scale up to +100k sites
- Availability close to 99.999 %
- Of course, high performance at lowest possible cost
- Need to control site creation from external app
- Automated and not-disruptive deployments
- Ability to execute drush commands on all sites
- Migration from previous platform
- Three devs only - frontend, backend and devops

What??

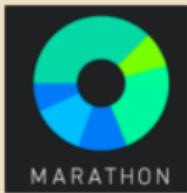


Seems scary, uh?

- Usual technologies are not enough to tackle project
- We are “computer scientists”, aren’t we?
- God bless Open Source
- Stairway to heaven cloud
- Docker is more than local

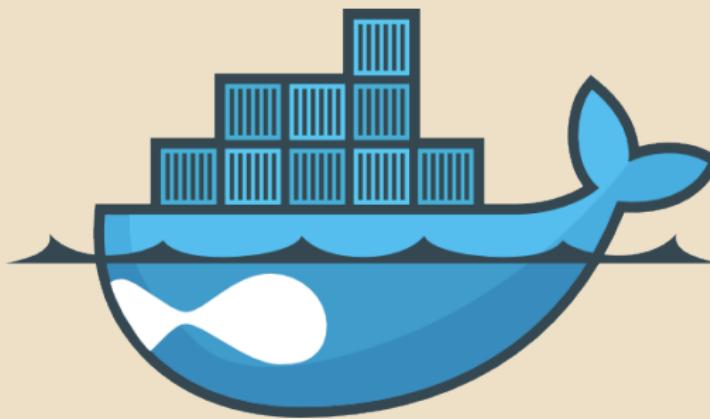


...funny times!



Drupal over Nginx and PHP-fpm

- Nginx allow a real very flexible configuration
- PHP fpm increases performance significantly
- We use DynamoDB to identify existent sites
- Mod Security with custom rules for Drupal - soon!



The Node.js app

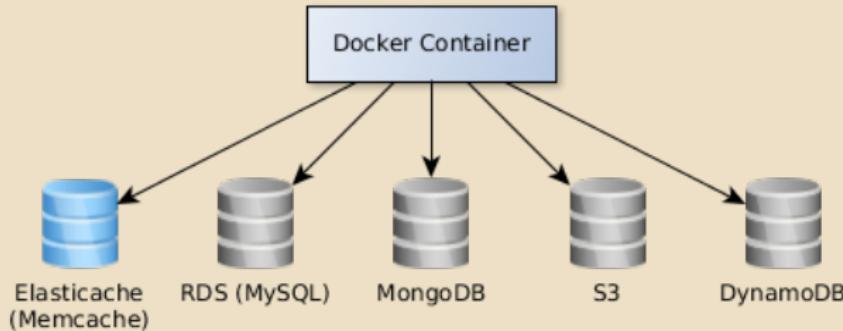
Manage sites on a multisite Drupal

- Asynchronous: execute something, and forget about it
- Use DynamoDB to store a list of deployed sites
- API to allow batch operations on sites list



Stateless Drupal container

- Memcache, MySQL, MongoDB as external services
- S3fs module → all files on S3
- Emails sent through PostMark
- Monitorization using NewRelic



Apache Mesos and Marathon

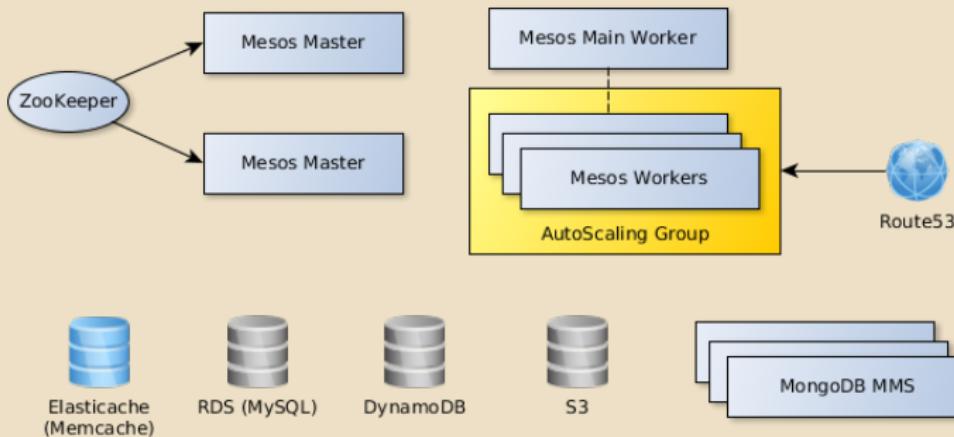
- Mesos abstracts the resources and creates a cluster
- Two masters orchestrated by Zookeeper, one or more workers
- Marathon allows to run Docker containers
- Chronos allows to run cron jobs
- Rest API to manage apps and containers



The AutoScaling Group

Mesos over AWS

- EC2 image for autoscaling from Mesos main worker
- AMI contains all 3 Docker images
- Main worker has Node.JS app and Varnish
- Script to sync amount of Marathon apps with workers number



And automatization for all ...

- Lazy DevOps is best DevOps
- Ansible let us manage AWS in a simple (and powerful!) way
- Makefile + Dockerfile = Over 9000!
- Create and destroy environments with a single command



**CAN'T SOMEONE ELSE
JUST DO IT?**

Devil is on details

- AWS do backups, but do not use default settings!
- Security groups for EC2 → external and internal firewall
- Remember to use protection → Http Auth for all Rest APIs
- Log centralization is a must → All hail Mesos!
- All of this does not matter without a recovery plan



Contents

1 Introduction

2 The Project

3 Problems and Solutions

4 Demo

5 Conclusions

The Database(s)

So you think 30k sites are too much

- 30k sites on a Drupal multisite means 30k databases
- MySQL → 1 folder per database, 1 file per table
- MongoDB preallocates files when a database is created
- Conclusion? Unmanageable



The Database(s)

Divide and conquer!

- Identify each site using a unique hash
- Use that hash as prefix for tables on MySQL and MongoDB
- Group 500 sites per database, both MongoDB and MySQL
- MongoDB maximum database size is 3.952GB
- MySQL number of tables per database is 62k aprox.



The Way Of The Request

Never seen a settings.php file with steroids?

- Connect to DynamoDB
- Identify domain from the request
(both drush and http)
- Calculate domain unique hash
- Ask DynamoDB for the database
where the site lives
- Make sure hash is used as prefix



The Site Creation Timeout

In soviet Russia, Nginx configures you

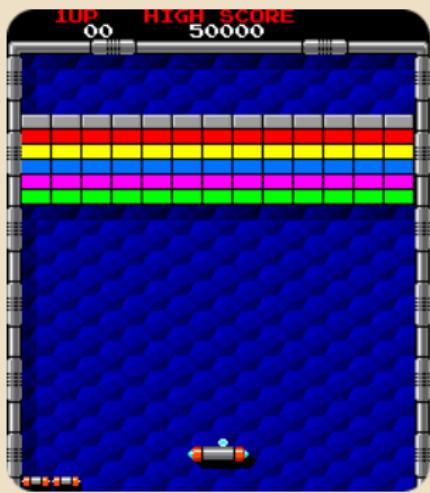
- Happened when Node.JS app launched a site creation
- Strange timeout error made us configure Nginx deeply
- Fire and forget. When site was created, app is notified
- Further investigation → timeout produced on Haproxy



Other problems found

Another brick in the wall

- Unstability on autoscaling group
- MongoDB MMS instances out of space
- PHP-fpm instability → slowlog
- Thousand S3 buckets? God bless S3fs
- Remote & massive Drush
- Migration: 30k sites on few days
- AWS RDS speed: the bigger, the better



Contents

1 Introduction

2 The Project

3 Problems and Solutions

4 Demo

5 Conclusions

Please, do not pull your arms out of the window

Demo time!

Contents

1 Introduction

2 The Project

3 Problems and Solutions

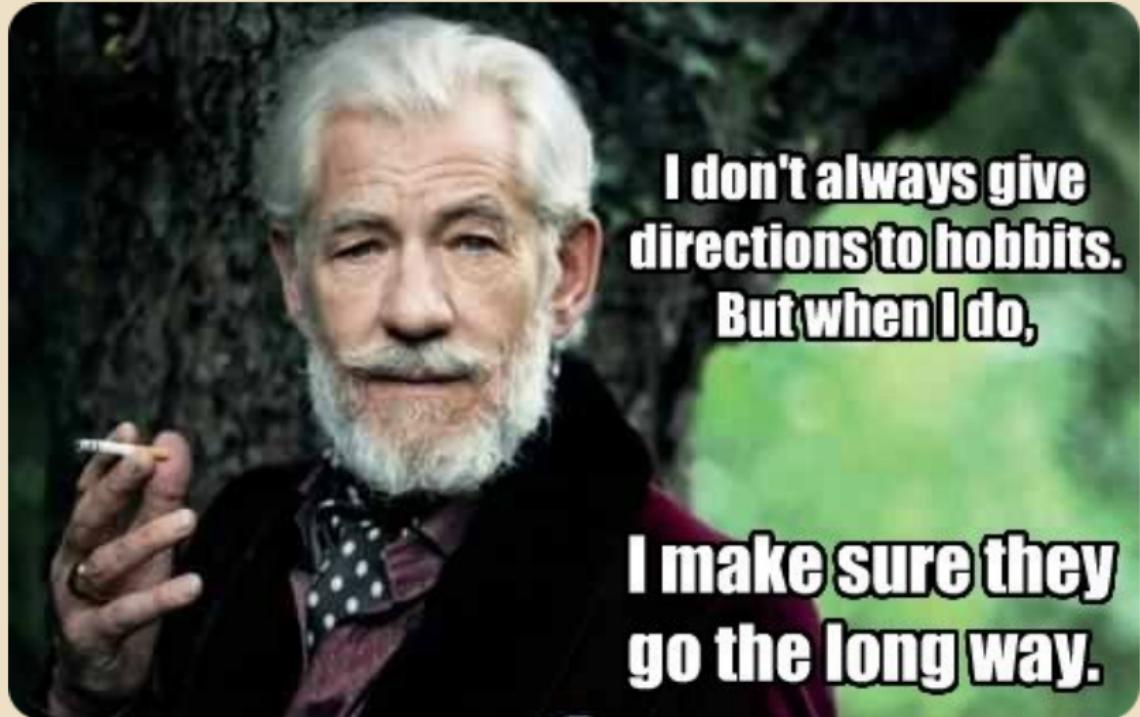
4 Demo

5 Conclusions

Spoiler alert!

- Project went live on April 2015. Absolute success!
- Learned a lot about new technologies
- Cloud + Containers = Profit
- Drupal flexibility is amazing

Be sure to practice first, Frodo



That's all, folks!

Thank you!

 @RabbitLair

 ezequielvazq[at]gmail[dot]com