Drupal in the Cloud

Scaling with Drupal and Amazon Web Services

Northern Virginia Drupal Meetup 3 Dec 2008

Cast of Characters

Eric at The Case Foundation: The Client

With typical client challenges

- Cost: Spending lots for boxes & load balancer in a data center
- Reliability

And some atypical ones

- Scale: Runs big campaigns -- say 48,000 people donating
  - 6-month grant competition http://miyo.casefoundation.org and
  - Or a 6-week challenge http://www.parade.com/contests/givingchallenge
- Boss likes to send out press releases

Frank at *Phase2 Technology*: The Architect

Dealing with Deployment Scenarios

- How many front/back end servers
- What to do about redundancy
- Automating the deployment process
- Aggressive timeline

Hurdles

- No significant deployments on AWS
- Building site AND deployment scripts
- Juggling the constellation of services, scripts, servers

Drupal: The CMS

We've done this for Drupal 5 and 6

AWS: The Cloud

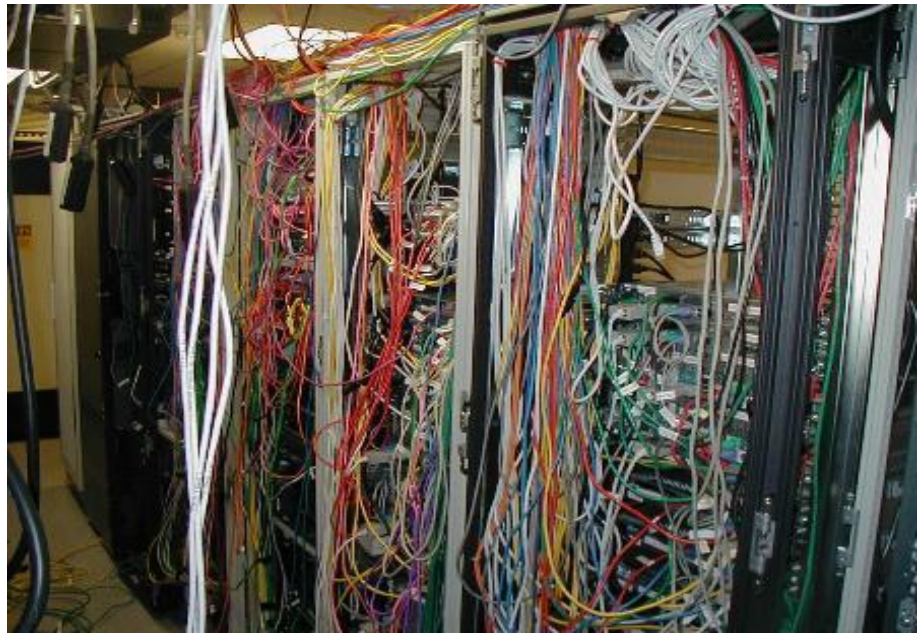
Collection of infrastructure services provided by Amazon

- S3 - Simple Storage Service
  - Scalable http read-write storage
  - Slow but bulletproof, pay per GB stored and transfered
- EC2 - Elastic Compute Cloud
  - Web service that provides resizable computing capacity
  - Pay by size of instance, time running, transfer rates
- EBS - Elastic Block Storage for EC2 data
  - Fast, perminent until the datacenter burns
- CloudFront: Content delivery network
- SimpleDB: Non-relational db
- Simple Queue Service: A message queue for splitting jobs among machines

How does S3 work?

- Upload into buckets via REST/SOAP
- Download like a webserver via HTTP
  - http://s2.amazonaws.com/basec/171890/2329706/copytitle.jpg
- Decentralized and fault tolerant

How does EC2 work?



- Virtualized servers across Amazon hardware infrastructure
- Commands are executed via REST/SOAP
- You can use front end services to ease the pain (RightScale)
- Pick your instance size
- Pick your OS
- Build an Amazon Machine Image (AMI) or use a stock/contributed AMI
  - AMI is a snapshot of a server installation, configured packages, etc.

Act 1

How do I save my stuff?

The Cloud is not like Kansas

EC2 servers are virtual

- They can die at any time
- Disk goes poof
- You don't even know where they are
- You can add more whenever load (or whim) dictates

This means...

- Configuration should be fast/automatic
- Server monitoring, please
- Load balancing, too
- And how do I save my stuff?

So how to save your stuff -- if you're Google

1. Non-relational database like SimpleDB (or BigTable)
    - No tables, just key-value pairs -- a giant hash
    - Data is safe, but not necessarily consistent
    - No locking, so scales really well
    - This is how Google does it
2. But Drupal is tied tightly to the DB
    - So this doesn't work
    - http://buytaert.net/drupal-in-the-cloud

How to save your stuff -- if you're Drupal

For the Database:

1. Use EBS for persistent storage of db disk
2. Use master-slave MySQL & backup
    - A vendor like RightScale can help with setup

For the Files:

1. Write files to S3
2. Or write them to EBS

Act 2

Configuration

Deployment Architecture

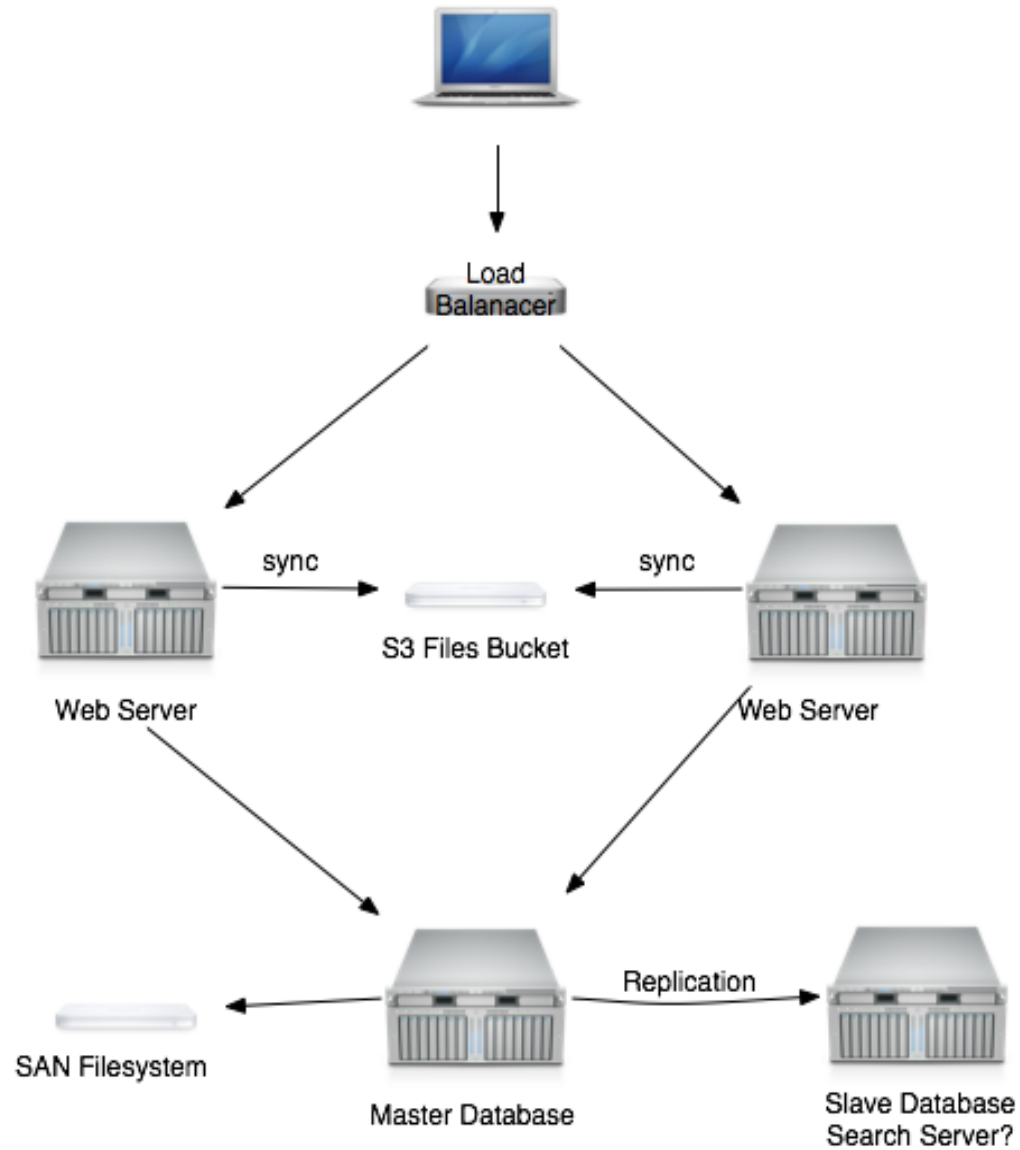
- You must know your deployment architecture before you can build it and (more importantly) scale it.
- Build for your worst (best?) case scenario
- Have a scaling strategy & implement acordingly

Single-tier Deployment

- Very easy to setup
- Difficult to scale

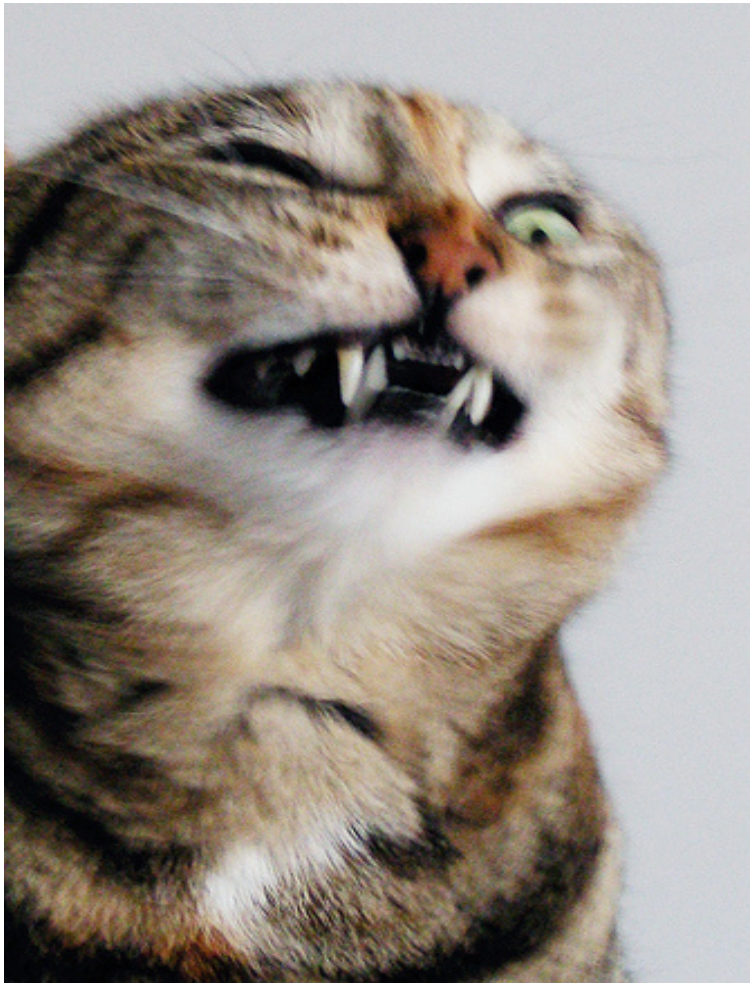
Multi-tier Deployment

- More difficult to setup
- Scales easily



Did someone say Pirates?!?!?!?!

Arrrrrrrrrrr!



Grrrrrrrrrrr!

Getting this show on the road

- Use EC2 command line, ElasticFox, or a provider such as RightScale
- Launch your instance
- Configure your instances
  - Via shell scripts, RightScripts
  - Assign Elastic IP or register dynamic with DNSME
  - Mount ESB filesystem(s)
  - Restore backups/snapshots as necessary
  - Configure webserver
  - Deploy Drupal from SVN or S3 (if needed)
  - Install cron jobs
  - Start services

Sure but HOW?!?!?!?

Using Plain EC2

- Largely manual process
- Write shell scripts to configure server (bash, ruby, perl, etc.)
  - Make all variations environment variables
- Create packages of scripts, script runners, and naming conventions to order scripts
- Upload scripts to S3 bucket or commit to SVN
- Start an AMI
- Post AMI boot
  - Setup environment variables for the instances
  - Get scripts on server from S3 or SVN
  - Run scripts to configure the server

Wait a minute, Eric says...

Better living thru....

RightScale to the rescue

- More expensive, but great framework to manage your setup
- Automates a great deal
- Create RightScripts
- Create Alerts & Escalations
- Build Server Templates
  - Assign AMI
  - Assign RightScripts - Boot/Operational/Decommission
  - Provide Inputs (env vars)
  - Assign Alerts/Escalations
- Bulk of the work is at design time
- Runtime deployments/scaling are button clicks (or automated!)

Disaster Planning

It will never happen to me

- Plan for disaster and test for it
  - Kill Web or Database instance
  - See if you survive and make fixes until you do
- Things that can help you survive
  - Master-Slave DB Replication
    - Daily S3 backup for Master
    - 10 minute S3 incremental for Slave
  - Filesystem sync to S3
  - Even better, ESB for filesystems & ESB to S3 backup
- Test recovery from complete failure of all instances

Other Cloud Providers

- Rackspace/Mosso
  - http://mosso.com

- Scalr/EC2
  - http://scalr.net

- Google App Engine (Python Only)
  - http://code.google.com/appengine/

- Coming Soon to EC2
  - Web Based Console
  - Built-in Load Balancing
  - Auto-Scaling options
  - Monitoring

Act 3

Cost and other stats

After all that, Eric saved some money

- Hosting expenses now 25% of what they were
- Cost now varies with what we use
- No more year-long hosting agreements
- Rolling out additional servers is a 15-minute job

Questions?

Or drop us a note...

Eric Johnson
Director of Web Development
The Case Foundation
ericj@casefoundation.org

Frank Febbraro
CTO
Phase2 Technology
frank@phase2technology.com

Slides will be posted later tonight

- Eric's blog: http://el-studio.com/tag/cloud/
- Phase2 blog: http://agileapproach.com