

# Modern Web development and operations practices

Grig Gheorghiu  
VP Tech Operations  
Nasty Gal Inc.  
@griggheo

# Modern Web stack

- Aim for horizontal scalability!
- Ruby/Python front-end servers (Sinatra/Padrino, Flask) with heavy-duty JavaScript libraries (jQuery, bootstrap)
- Intermediate RESTful API layer (Jetty/Groovy)
- Back-end services (e-commerce platform, payment gateway)
- Indexing/search services: Solr, ElasticSearch
- Database layer (relational such as MySQL/ PostgreSQL, noSQL such as Riak/Cassandra)

# Operations stack

- Hypervisor: KVM
- Private/Hybrid cloud: OpenStack, CloudStack, Eucalyptus
- Public cloud: Amazon, GCE, Rackspace, Joyent
- OS: Ubuntu 12.04 LTS
- HTTP proxy/SSL terminator: nginx
- Load balancer: haproxy
- CDN: Akamai, EdgeCast, Fastly, Insight Logic
- DDOS protection: CloudFlare

# Development and deployment pipeline

- Github or private Git install
- Deployment done with Jenkins and Capistrano/Fabric
- Devs commit and push code which triggers Jenkins run in development environment (Continuous Integration)
  - Use Vagrant if you can so dev and prod are on the same OS
  - Unit tests get run
- QA team deploys to staging environment
  - Integration tests get run including Selenium tests
- Ops team deploys to production environment

# Monitoring, graphing, logging

- **If it's not monitored, it's not in production**
- **Monitoring is for Ops what testing is for Dev**
- External monitoring: Pingdom, Keynote
- Internal monitoring tools: old (Nagios) and new (Sensu)
- Graphing tools: Graphite, Cacti, Ganglia, Munin
- Logging tools: Logstash, ElasticSearch, Kibana (or go commercial: Splunk)
- Alerting tools: PagerDuty

# Dashboards!



# Configuration management

- Indispensable at scale
- Infrastructure-as-code
- Chef or Puppet, take your pick
- We use Chef
  - Wide choice of community 'cookbooks' for major packages (Apache, nginx, haproxy, mySQL etc)
  - Decent development pipeline (cookbooks pinned to versions in different environments)
  - Decent testing tools: Foodcritic (lint), Chef Spec (unit testing) Test Kitchen (integration testing using Vagrant)

# What is 'devops'?

- Open lines of communication and strong collaboration between Dev, QA and Ops
  - JIRA: ticketing mechanism keeping everybody in the loop
  - Change Management Requests (CMR) for production changes
  - Jabber for communication ('operations' chat room)
- Devs need Ops skills; Ops need scripting skills at a minimum; both need testing and DBA skills ('full stack')
- Automation! It's the cloud era
- One of the goals of a devops team is to deploy to production N times a day with no downtime



# To cloud or not to cloud

- Public cloud
  - Advantages: fast provisioning, 'infinite' elasticity
  - Disadvantage: poor reliability, poor performance in some cases ('noisy neighbors')
- Private cloud
  - Advantages: tunable performance, increased control, potentially better security
  - Disadvantage: decreased elasticity unless you are Facebook
- Hybrid cloud: best of both worlds
  - Start with private cloud but be ready to expand horizontally into public cloud

# Do you need Big Data?

- Only if the volume of data exceeds a big bare-metal box running MySQL or PostgreSQL
- Most people don't need Big Data
- If you do, try Hadoop/Hive/Pig
  - Export data to S3 in csv.gz format
  - Read it in Hive tables
  - Run Hive/Pig queries, export to MySQL
  - Build dashboards on top of MySQL

# Architecture is important

- Horizontal layers that can be scaled independently
  - Both software and hardware should scale horizontally
  - Take advantage of The Cloud
  - DB layer (hard to scale horizontally if relational, easier if NoSQL)
  - RESTful API layer
  - Front-end layer – responsive design, no difference between mobile and desktop
  - Software load balancers in High Availability mode (haproxy + keepalived)
- No single points of failure!

# Troubleshooting skills are important

- Interview question: talk about all the different layers (both software and networking) that are involved in a user typing [www.google.com](http://www.google.com) in a browser and getting the Google home page back
- What can go wrong at each layer?
- Need to understand the major Internet protocols
  - TCP/IP (tcpdump)
  - DNS (dig, nslookup)
  - HTTP (curl, Firebug, twill, Selenium)
  - SMTP