Getting to Elastic

Adapting a Legacy Vertical Application Environment for Scalability

Eric Shamow - USENIX LISA 2011

Who Am I?

- Professional Services Engineer for Puppet Labs
- 13 years in IT, from line SA to manager of large Operations group
- eric@puppetlabs.com
- @eshamow
- http://www.opsrealist.com

Overview

- What's at Issue?
- Characterizing the Problem
- Resolving the Problem
    - Culture
    - Metrics
    - Infrastructure
- Tying it all together

What's at Issue?

- What does "elastic" mean?
- Increasing drive to move to "the Cloud"
- What does elastic mean in our new environment?

What's at Issue?

- What does "elastic" mean?
- Increasing drive to move to ~~"the Cloud"~~ IaaS/PaaS/SaaS
- What does elastic mean in our new environment?

Characterizing the Problem

Five Questions

- How are servers deployed?
- Can our apps handle it?
- When should we expand?
- When do we contract?
- What tooling do we use?

How Are Servers Deployed?

How Are Servers Deployed?

Elasticity Requires Automation

When Should We Expand?
When Should We Contract?

When Should We Expand?
When Should We Contract?

Elasticity Requires Open Metrics

When Should We Expand?
When Should We Contract?

Elasticity Requires Open Metrics

Open Metrics Require Culture Change

What Does This Have to Do With DevOps?

- Be a professional
- Share and collaborate on information and resources
- Build trust
- Consider surrendering unilaterally
- Compromise

Why is Communication Essential?

- Operations does not understand the application
- Development does not understand the environment
- Everyone assumes that their understanding of the problem is complete

Metrics Through Collaboration

- Provide Operational Logs to developers
- Request structured data in logs developers provide to Operations
- Key question: what are the pain points for each group?
- Monitor everything, but don’t focus on what doesn’t matter until you’ve managed the rest

Can Our Apps Handle It?

- Scale Out, Not Up
- Latency in “the cloud” is not high or low – it’s variable
- What is your ROI on additional nodes?
  - Linear
  - Logarithmic

Be Prepared for Change

Be Prepared for Change

Bottlenecks will move inside your stack

Be Prepared for Change

Bottlenecks will move inside your stack

Have a process for accepting and handling the changes

When Should We Expand? When Do We Contract?

- Blind automation can be dangerous
- Impose sanity limits on builds and teardowns
  - How many can I provision/destroy?
  - How fast can I provision/destroy?
- Alert humans in edge cases
- Consider a pool of offline servers to speed operation

puppet labs

Tooling (The Short Version)

- Network
  - DHCP, PXE, DNS, OS and Patch provisioning all must have APIs or script-based management
- OS
  - Cobbler, Spacewalk, Foreman
- Configuration Management
  - Puppet! (but seriously, please use something)

Tying It All Together

- You’ve already done the hard work
- Servers can be provisioned based on
  - Metrics agreed upon by business, dev and ops
  - API-based tooling around infrastructure and automated deployment
- What next?
  - Ticketing and change control/review
  - Integration testing

Q & A