Cloud... so much more than a tools fest

Qcon London 2012
Vagrant & Veewee

Web Operations: Keeping the Data On Time

http://jedi.be/blog
Context: a traditional enterprise
With a developer group on a mission to innovate
Realtime TV - Web interaction
Second Screen Applications
ENTERPRISE CAT
IS READY FOR PROVISIONING
Maximum warp!

TV-show
+- 1 million concurrent viewers
ENTERPRISE CAT

IS NOT A SIGN

http://www.twylah.com/billycoover/topics/xamarin
Cloud Cat

Doin' his cloud thing
Abstraction is AWESOME
If it fails hard to debug/understand
Too generic

Java  
Python  
Nodejs  
Esper  
Redis  
Hadoop  
Mongodb  
Mysql  
ETL tools
Need more control
Oh, here's your problem...

IAAS

The computer is missing
I never did any cloud before. Only enterprise stuff
What I learned from managing Production Servers
A Single Server
Generic Servers become Specific Servers
Physical becomes Virtual

WEB

WEB

WEB

APP

APP

APP

DB

xen/vsphere/kvm/...
We learned cloning isn’t working
We introduce config management

WEB

= Config

WEB

= Config

WEB

= Config

WEB

= Config

APP

= Config

APP

= Config

APP

= Config

DB

= Config

DB

= Config

DB

= Config

JEOS

JEOS

JEOS

JEOS
Infrastructure as code

WEB
Config
JEOS

APP
Config
JEOS

DB
Config
JEOS

Config Management
Code Repository

Cfengine/Puppet/Chef
What I learned from managing Test Servers
Reuse Across Environments

Infrastructure
Code Repository

DEV

TEST

WEB
APP
DB

PROD

WEB
APP
DB
Explosion of VM creation
UI Interface to Automated Provisioning of VMS

Cobbler, Spacewalk, ...
Metadata Registry for Systems

Config Management

Puppet/Chef Server
we hearz thingz
from de other side...
You had beans, didn’t you
I’VE NEVER INSTALLED A GUI FOR THAT.
$ knife ec2 server create \n-r 'role[webserver]' -I ami-7000f019 \n-f m1.small \n-A 'Your AWS Access Key ID' \n-K 'Your AWS Secret Access Key'
$ puppet node create --image ami-XxXXXxXXX \
  --keypair puppetlabs.admin --type m1.small

aka “cloudpack”
AWS Network Constraints

Security
Groups only on creation

Only 1 network interface

ELB not on non-standard ports

Dynamic Monitoring

nodes = search(:node, "hostname:[* TO *] AND chef_environment:#{node.chef_environment}"")
VM creation failure, network hiccups, disk erratic behavior

the "up" button
push the "up" button

"UP"..."UP".....the "UP" button !
PUSH THE "UP" BUTTON.....
Embrace Fail.
Architect cat sez

Re-architect

bak to drawing bord for joo
“Quis custodiet ipsos Custodes”
who watches the watchers
What I learned working in the (Amazon)cloud
what's the difference?

<table>
<thead>
<tr>
<th>Internal</th>
<th>Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web,App,DB</td>
<td>Web,App,DB</td>
</tr>
<tr>
<td>Config Mgmt</td>
<td>Config Mgmt</td>
</tr>
<tr>
<td>Metadata Registry</td>
<td>Metadata Registry</td>
</tr>
<tr>
<td>JEOS</td>
<td>AMI</td>
</tr>
<tr>
<td>VM</td>
<td>Xen</td>
</tr>
<tr>
<td>UI Provision</td>
<td>AWS Console</td>
</tr>
</tbody>
</table>
From console to API

Web UI

AWS Console

“Internals” API

AWS API

Abstracted API

Fog
Jclouds
Boto
Beyond Servers Components

Server(s)

+ 

Loadbalancers
DNS Service
IP Address
Email Service
EBS Volume
Firewall
Keys
From server to **stack**

Cloudformation

Json file specifying order of component creation and dependencies
What I learned from managing development Servers
Development moves from host into **virtual machines**
Development starts using **config mgt**
reuse “code” across environments
Vagrant

UP
PROVISION
HALT
DESTROY

Simple CLI

Vagrantfile

VM Management

Puppet/Chef
Basebox

Config
JEOS
HOST

WEB
APP
DB

http://vagrantup.com/
Integrate with **Continuous Integration**

**DEV**
- Infrastructure
- Application Code Repository

**TEST**
- Infrastructure
- Application Code Repository

**PROD**
- Infrastructure
- Application Code Repository
Setup Outgrew VM on Laptop

Setup Outgrew Test Lab

Peak Capacity needed in Prod

CLOUD
Reuse “workflow” across Environments

“If it’s hard to it more often”
+Reuse workflow across hypervisors customers

- Web, App, DB
- Config Mgmt
- JEOS
- Virtualbox

- Web, App, DB
- Config Mgmt
- JEOS
- KVM

- Web, App, DB
- Config Mgmt
- JEOS
- AWS

- Fog Library
Cloud Libs (Jclouds/Fog/Boto) embracing **old and personal**

<table>
<thead>
<tr>
<th>“personal”</th>
<th>“old”</th>
<th>“hybrid”</th>
<th>“new”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualbox</td>
<td>Vsphere</td>
<td>Openstack</td>
<td>AWS</td>
</tr>
<tr>
<td>VM Fusion</td>
<td>Kvm</td>
<td>Eucalyptus</td>
<td>Rackspace</td>
</tr>
<tr>
<td></td>
<td>Libvirt</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Abstracting OS Installation

Debian  Ubuntu  Solaris  Win  Redhat
Archlinux  Suse  Centos

Kickstart  Preseed
Unattended.xml

Virtualbox  KVM  Fusion  Parallels

http://github.com/jedi4ever/veewee
Cloud Libs (Jclouds/Fog/Boto) beyond servers

DNS    Keys    IPs    Storage
Securitygroups    Loadbalancer
Workflow **beyond servers**

VM

up
provision
halt
destroy

Other

ip
balance
sorry
Self Servicing
Continuous Integration to Continuous Delivery

Faster/Delivery

DEV  TEST  PROD

Faster/Feedback

Infrastructure Code Repository

Application Code Repository
Confidence
Faster/Delivery

DEV  →  OPS

Faster/Feedback
Infrastructure ~ Code
TDD Cycle

Add Test

Refactor

Run tests

Write Code

Watch Test Fail
Testing ~ Monitoring
Debugging ~ Metrics
Metrics Reuse

Traditional Ops

TEST

WEB  APP  DB
WEB  APP  DB
WEB  APP  DB

PROD

WEB  APP  DB
WEB  APP  DB
WEB  APP  DB

Collectd, Ganglia, Graphite, Opentsdb
Extend “metrics” to Development

Collectd, Ganglia, Graphite, Opentsdb
Extend “logs” to Development

Logstash, Graylog
Selfservicing “metrics injection”
Selfservicing “alerts”
Selfservicing “graphs”
Repeating Service Pattern
Simple API/CLI, Self-Servicing
Reuse workflow across monitoring tools

Collectd
Nagios

Ganglia
Zenoss

Graphite
Sensu

“my dream” Library
Workflow reuse

Monitoring UP
Abstracting

It's all events

Timestamp - Key - Value

Logs  Metrics  Monitoring  Meta Ops
Social IT
radiate information to where it’s needed

Business ‘Pulse’
So maybe you don’t have unlimited resources
But you can shape your internal IT as a cloud
Thank you!