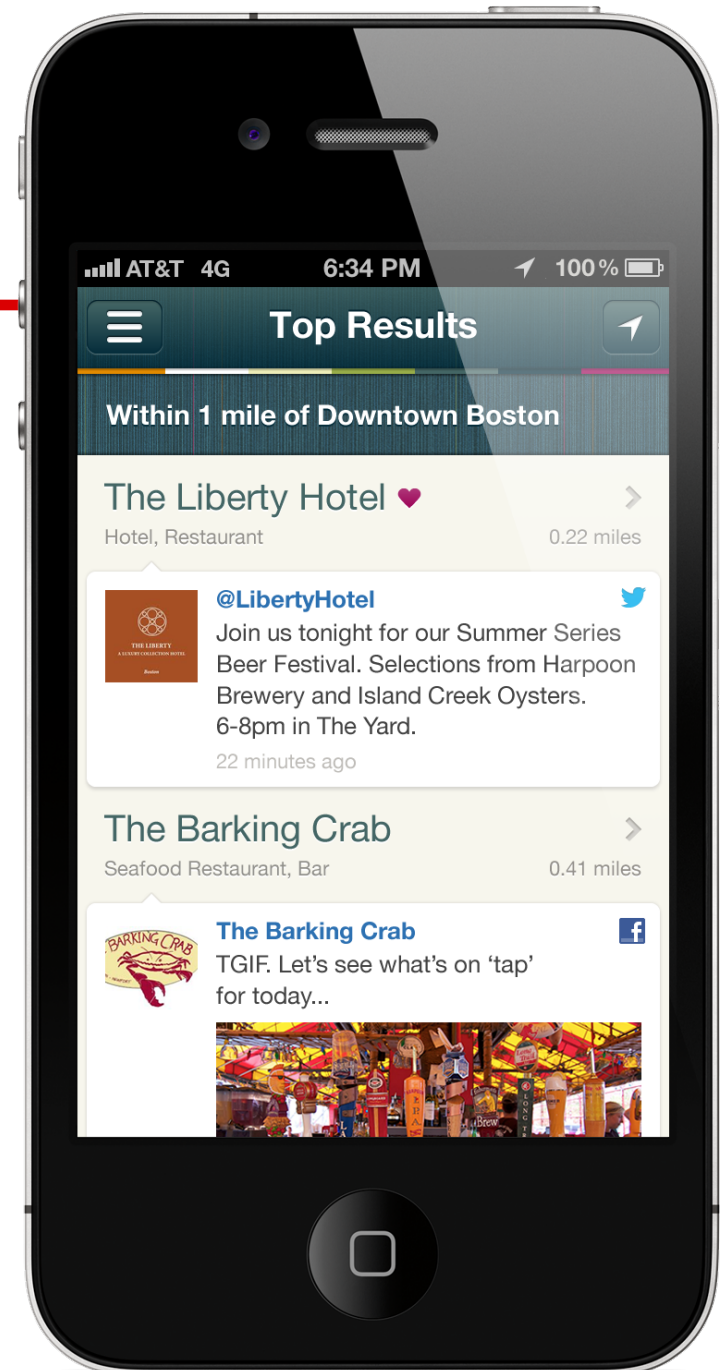

Otto: unifying application & platform management

Alex Lambert (@alambert)
Co-founder & CTO
alex@spindle.com



The problem

- Deploy 10+ applications
 - Scala: content fetching
 - REST API
 - Web
 - ElasticSearch
- Frequent deployments
- Dynamic infrastructure
- Seed-stage startup



Wishful thinking

- Simple & transparent
 - Low maintenance
 - Easy version bump
 - Easy machine enlistment
 - Retry after failure (cloud)
 - Technology-independent
 - Safety
 - Audit & change notification
-

From deployment to management

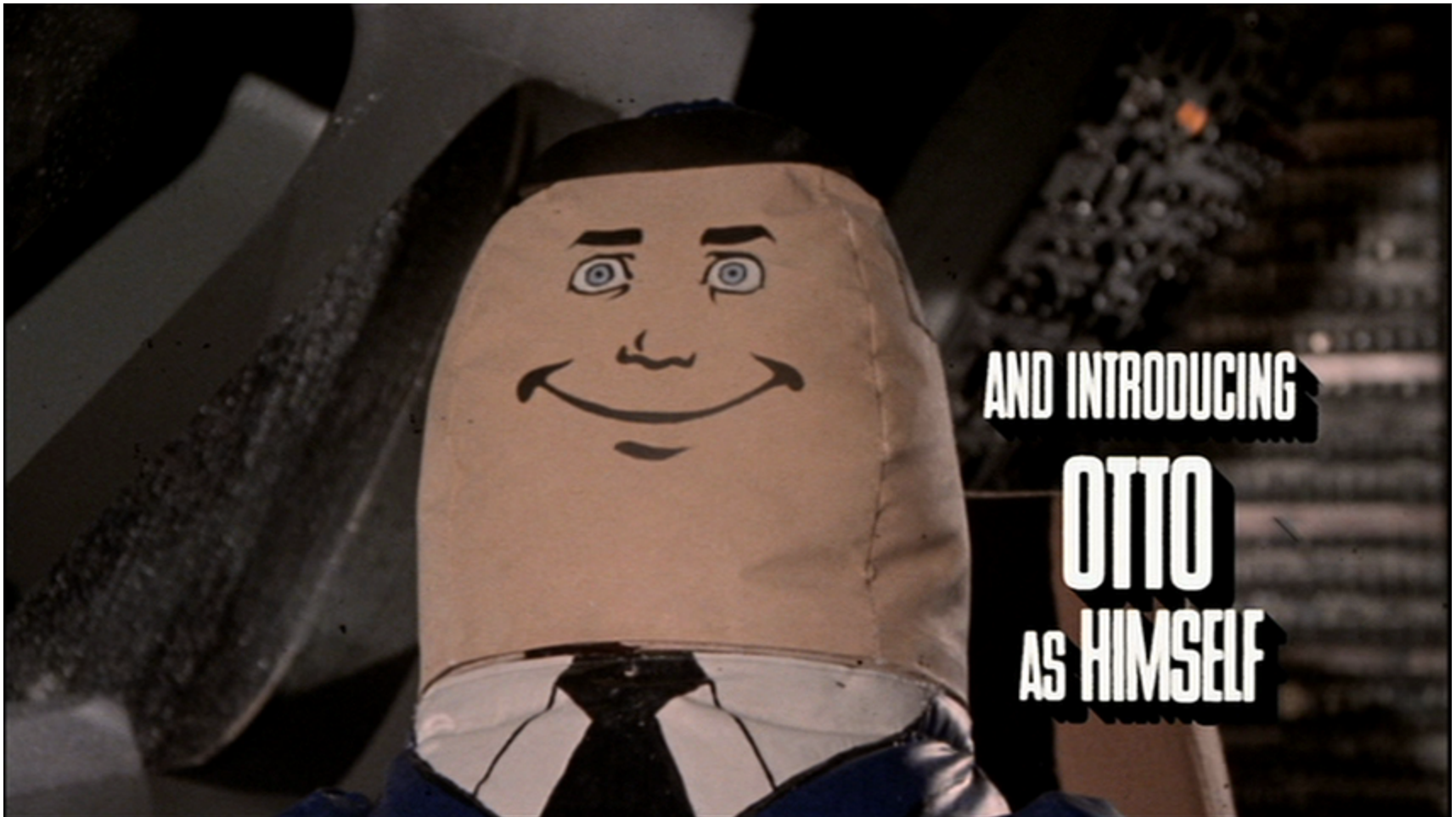
- Easy separation of privilege
 - Prevent app configuration drift
 - Partial & rolling app deployments
 - Start app on machine reboot
 - Restart app on failure
 - Configuration-only changes
 - Work with build server
 - Deploy the latest daily build at 3am
 - Don't restart everything at the same time
-

From *application* to *platform*

- Apps have platform dependencies
- *Platform*: everything except our code
 - Java (& java.policy)
 - nginx, SSL, static assets
 - rsyslog for remote logging
 - Software RAID

Where does the platform end & the app begin?

Can we manage them together?



Otto unifies app & platform management
(in 200 lines of code)

Introducing Otto

We've solved these problems for the *platform*

Puppet: solid platform management

1. *Declare* the desired state of your system
 - Resources (the package "nginx")
 - Attributes ("1.2.4 is installed")
 - Dependencies ("install SSL certs before vhosts")
2. Puppet *enforces* the desired state
3. PROFIT!!

Otto extends Puppet to manage *applications*

A natural resource

```
user { 'dave':  
  ensure      => present,  
  uid          => '507',  
  gid          => 'admin',  
  shell        => '/bin/zsh',  
  home         => '/home/dave',  
  managehome  => true,  
}  
# puppet apply dave.pp  
notice: /User[dave]/ensure: created
```

A resource for an application

```
node "aws-w2b-indexer-2.spindle.com" {  
  class { "grok::app::indexer":  
    jenkinsProjectName => "master-checkin",  
    jenkinsBuildID => "104"  
  }  
}
```

The screenshot displays the Jenkins web interface for a specific build. At the top, the Jenkins logo and a search bar are visible, along with user links for 'admin' and 'log out'. The breadcrumb navigation shows 'Jenkins > grok-master-checkin > #104'. A sidebar on the left contains various links: 'Back to Project', 'Status', 'Changes', 'Console Output [raw]', 'Edit Build Information', 'Polling Log', 'Get Build Data', 'No Tags', 'See Fingerprints', and 'Previous Build'. The main content area features a large blue sphere icon and the text 'Build #104 (Nov 1, 2012 3:33:51 PM)'. To the right of this, it states 'Started 12 hr ago' and 'Took 15 min on master', with a button to 'add description'. Below this, the 'Build Artifacts' section lists several files, including 'indexer-assembly-0.1-SNAPSHOT.jar'. The 'Changes' section shows a single change: '1. LSI-72: fix error logging (detail)'. At the bottom, it indicates the build was 'Started by an SCM change (17 times)' and provides the 'Revision: 0d1729e8400c249335c98eee4a2773ead46bd5fd' with a link to 'origin/master'. A watermark of a cartoon character is visible in the background. The footer includes a link to 'Help us localize this page', the page generation time 'Nov 2, 2012 4:01:32 AM', and the version 'Jenkins ver. 1.466.2'.

Puppet: solid platform management

- Already used for infrastructure
 - Comprehensive built-in resources
 - files, services, mount points, packages, cron jobs
 - Simple DSL
 - Hides OS-specific details
 - `service { "nginx": ensure => "running" }`
 - Powerful
 - Popular, documented, supported
 - Declarative, not imperative: enforced model
-

Solid patterns and practices

- Puppet configuration: flat text files
 - Store them in git
 - Simple & transparent
 - Change notification
 - Auditing
 - (Optionally) serverless
- Amazon EC2: cloud-init

```
#cloud-config
puppet:
  conf:
    agent:
      server: "puppet.spindle.com"
  ca_cert: |
    -----BEGIN CERTIFICATE-----
    MIICUTCCAbqgAwIBAgIBAT....
```

Manage the platform *and* the app

Static HTML,
CSS

nginx
configuration

SSL cert

Java app

prod only:

rsyslog
forwarding

```
class grok::app::apollo($jenkinsProjectName, $jenkinsBuildID) {  
  file { ["/usr/share/nginx/spindle.com":  
    source => "puppet:///modules/grok/app/apollo/web/spindle.com", ...  
    require => Package["nginx-full"]  
  }  
  nginx2::site { "apollo":  
    configuration => template("../apollo/nginx/apollo.erb"),  
    require => File["/usr/share/nginx/spindle.com"], ...  
  }  
  nginx2::ssl_certificate { "apollo": ...  
  }  
  grok::otto_java_app_from_build { "apollo":  
    jenkinsProjectName => $jenkinsProjectName,  
    jenkinsBuildID => $jenkinsBuildID, ...  
    require => [Nginx2::Site["apollo"], Nginx2::Ssl_certificate["apollo"]]  
  }  
  if $environmentName == "app-prod" {  
    rsyslog::logentries_app { "apollo":  
      localSyslogPort => 5140,  
      logentriesToken => "a7e4255c-1234-5678-9012-8eda9eb3567c"  
    }  
    Rsyslog::Logentries_app["apollo"] -> Grok::Otto_java_app_from_build["apollo"]  
  }  
}
```

Deploying a new version

```
class { "grok::app::indexer":  
  jenkinsProjectName => "master-checkin",  
  jenkinsBuildID => "104" "105"  
}
```

- Change "104" somehow
 - Text editor
 - Or extlookup() with node name, facts...
 - Or generate()...
 - Or external node classifier knowing *global* state...
 - Machines sync every 30 minutes (+ skew)
∴ Partial deployment, continuous deployment,
automatic rollback
-

Deploying new configuration

- Just flat files deployed from puppet fileservers
- Safe replacement
- Automatic service restart

∴ Configuration drift eliminated

All the old showstoppers

What if the build server was down?

Puppet fetches & applies new policy every 30m

∴ Failures handled automatically

Now you're thinking with models

What if there's a new machine?

Puppet fetches & applies new policy every 30m

∴ New machines handled automatically

Demo

```
sudo apt-get install puppet
```

```
# public release coming soon!
```

```
git clone https://github.com/spindlelabs/otto.  
git  
cd otto
```

```
sudo puppet apply --modulepath modules:  
examples/helloworld/modules --debug  
examples/helloworld/manifests/site.pp
```

```
∴ helloworld installed, running in /opt/otto
```

```
otto::app { $appName:
  appBuildID => $appBuildID,
  appUserGroupName => $appUserGroupName,
  appBuildArtifactFetchCommand => sprintf("sh -c 'umask 077 && wget -O %s %s && chgrp %s %s && chmod
0640 %s && mv %s %s'",
                                          shellquote($appBuildArtifactTempName),
                                          shellquote($appBuildArtifactUrl),
                                          shellquote($appUserGroupName),
                                          shellquote($appBuildArtifactTempName),
                                          shellquote($appBuildArtifactTempName),
                                          shellquote($appBuildArtifactTempName),
                                          shellquote($appBuildArtifactName)),
  appBuildArtifactName => $appBuildArtifactName,
  appConfSource => $appConfSource,
  appRunContent => $appRunContent,
  require => [Class["java"], User[$appUserGroupName]]
}
```

Related work

- Etsy Deployinator
 - Great, but "...we have a bunch of directories...rsync to a staging area on each web box."
 - Fabric
 - "library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks."
 - Capistrano
 - Windows Azure
 - Native OS packages
 - LinkedIn Glu
 - Not platform management
-

Next steps

```
$ find modules/otto -type f | xargs wc -l  
153 modules/otto/manifests/app.pp  
48 modules/otto/manifests/init.pp  
201 total
```

OSS release in the next few weeks

<https://github.com/spindlelabs/otto>

alex@spindle.com, @alambert

