Otto:
unifying application & platform management

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

Wishful thinking

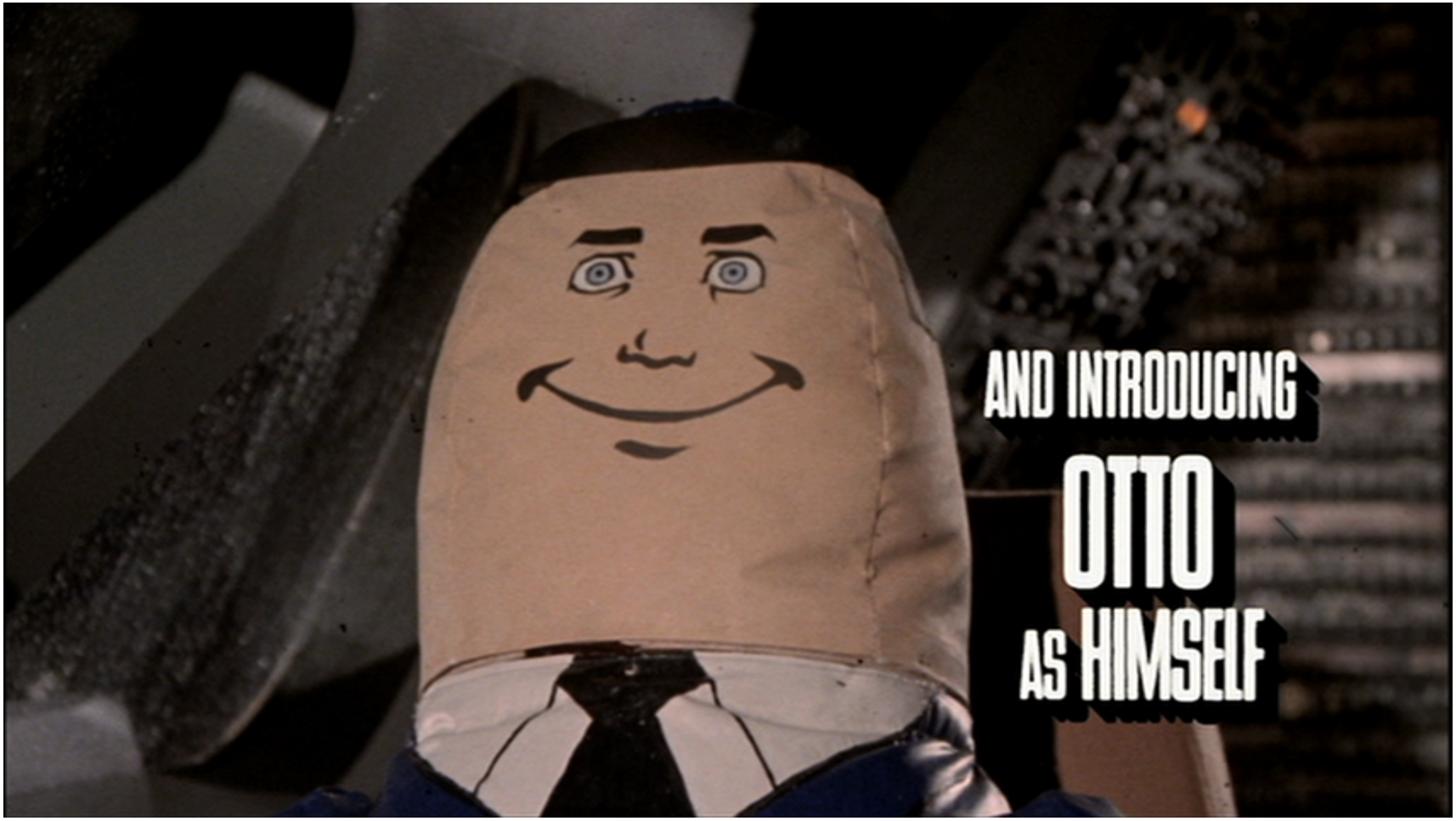
- Easy app version & configuration change
- Easy machine bootstrapping
- Restart app on failure & machine reboot
- Embrace failure

From *application* to *platform*

- Apps have platform dependencies
- *Platform*: everything except our code
  - Java
  - nginx configuration, SSL certs, static assets
  - syslog
  - 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

Puppet: solid platform management

1. *Model* the desired state of your platform
   - Resources (the package "nginx")
   - Attributes ("installed with version 1.2.4")
   - Dependencies ("install SSL certs before nginx")
2. Puppet *enforces* your model
3. PROFIT!!

Otto extends Puppet to manage *applications*

Models

```
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
```

change Dave's shell to bash

```
> puppet apply dave.pp
```

Dave's shell is now zsh

with Otto:

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

```
> puppet apply indexer.pp
```

Machine now runs build 104 from the master branch of the indexer application (installed or upgraded)

Start me up

Ubuntu cloud-init: EC2 user-data

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

Manage the platform *and* the app

static HTML, CSS

nginx configuration

SSL cert

JVM app

prod-only syslog configuration

```
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" {
    syslog::forward { "apollo":
      localSyslogPort => 5140
    }
    Syslog::Forward["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, `extlookup()`, `generate()`, external node classifier)

Puppet fetches & applies model every 30m

∴ Partial deployment, continuous deployment, automatic rollback

Now you're thinking with models

How do we add a service to a machine?

What if the build server was down?

What if someone changes config locally?

Puppet fetches & applies new policy every 30m

∴ New machines handled automatically

∴ Failures handled automatically

∴ Configuration drift eliminated

Next steps

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

More slides: https://spindle.com/talks

Preview: https://github.com/spindlelabs/otto

Formal release soon

alex@spindle.com, @alambert