

# Xconf Sydney 2014

Continuous Delivery  
*Provisioning*

**THE BORING STUFF**

# Antipatterns...

According to Jez Humble and Dave Farley:

- Deploying Software Manually
- Deploying to a Production-like Environment Only after Development Is Complete
- Manual Configuration Management of Production Environments

# Antipatterns...

According to Jez Humble and Dave Farley:

- Deploying Software Manually
- Deploying to a **Production-like Environment**  
Only after Development Is Complete
- Manual **Configuration Management** of  
Production Environments

**VAGRANT**

# Vagrant

- Create a directory and run `vagrant init` in it
- Copy `puppet.box` to the same directory and edit `Vagrantfile` to use this box
- Forward port 10080 on host to 80 on guest
- Run `vagrant up`
- Run `vagrant ssh`

# **PUPPET BASICS**

# Puppet - Scaffolding

- Uncomment the puppet section in Vagrantfile
- Create the `manifests` directory and create the `site.pp` file inside it
- Run `vagrant provision` to apply the puppet manifests



# Puppet - Basics

## Installing packages

- Edit `site.pp` to ensure `tree` and `httpd` are installed

## Turn off the firewall

- Edit `site.pp` to ensure the `iptables` and `ip6tables` services are not enabled or running

<https://docs.puppetlabs.com/references/latest/type.html>

...or just google “puppet type reference”

# Puppet - Basics

## Run an arbitrary command

- Untar the sample tarball into `/usr/local`

Download the tarball first:

```
curl -L http://bit.ly/1pNAr3t > sample.tar.gz
```

## Place a file on to the file system

- Put the following content into `/etc/profile.d/sample.sh`

```
export PATH=/usr/local/sample/bin:${PATH}
```

**PUPPET CLASS**

# Class *Definition*

```
class junk {  
  file { '/tmp/junk.txt':  
    ensure => file,  
    content => 'Trash some rubbish',  
  }  
}
```

# Class *Declaration*

```
class { 'junk': }  
  
...  
class { 'junk': }
```

**Error: Duplicate declaration: Class[Junk] is already declared**

# Class *Declaration*

```
include junk
```

```
...
```

```
include junk
```

**Notice: Finished catalog run in 0.60 seconds**

# Defining Parameterized Class

```
class junk($verb, $name = 'rubbish') {  
  file { '/tmp/junk.txt':  
    ensure => file,  
    content => "${verb} some ${name}",  
  }  
}
```

# Using Parameterized Class

```
class { 'junk': verb = 'Trash', name = 'rubbish' }
```

Gives you "Trash some rubbish"

```
class { 'junk': verb = 'Dump', name = 'garbage' }
```

Gives you "Dump some garbage"

Mandatory parameter `verb` – cannot use the "include" syntax



# Create a *lamp\_stack* class

## Packages:

- httpd, mysql-server, mysql, php, php-mysql

## Services

- httpd, mysqld

## File /var/www/html/test.php

- Content:

```
<html><body><?php echo "We have lift off!"; ?></body></html>
```

# **PUPPET MODULES**

# Module Structure

- Module Name
  - manifests
  - files
  - templates
  - ...

[https://docs.puppetlabs.com/puppet/latest/reference/modules\\_fundamentals.html](https://docs.puppetlabs.com/puppet/latest/reference/modules_fundamentals.html)

# The lamp Module

- Move the previous `lamp_stack` class into the `lamp_stack` module
- Put the content of the file into a file (or template)
- Declare a `$user` variable in the class
- Create another file (`hello.html`) from a template (`hello.html.erb`)

```
<html><body>Hello <%= @user %></body></html>
```

# Configure puppet

- Vagrantfile

```
config.vm.provision "puppet" do |puppet|  
  puppet.manifests_path = "manifests"  
  puppet.manifest_file  = "site.pp"  
  puppet.module_path = ['modules']  
end
```

# **EXTERNALISE CONFIGURATION USING HIERA**

# Configure puppet

- Vagrantfile

```
config.vm.provision "puppet" do |puppet|  
  puppet.manifests_path = "manifests"  
  puppet.manifest_file  = "site.pp"  
  puppet.module_path = ['modules']  
  puppet.hiera_config_path = 'hiera.yaml'  
  puppet.working_directory = "/vagrant"  
end
```

# Configure hiera

- hiera.yaml

```
---
:hierarchy:
- {%environment}
- common
:backends:
- yaml
:yaml:
  :datadir: hiera_data
```



# Add hiera data

- Create a hiera\_data directory
- Create development.yaml

```
---
```

```
user: Karl
```

- Create common.yaml

```
---
```

```
user: Superman
```

# Doing the hiera lookup

- In the lamp\_stack class  
`$user= hiera('user')`

# Configure puppet

- Vagrantfile

```
config.vm.provision 'puppet' do |puppet|  
  puppet.manifests_path = 'manifests'  
  puppet.manifest_file  = 'site.pp'  
  puppet.module_path = ['modules']  
  puppet.hiera_config_path = 'hiera.yaml'  
  puppet.working_directory = '/vagrant'  
  puppet.options = '--environment development'  
end
```

# **EXTERNAL NODE CLASSIFICATION**

# Using hiera

- Comment everything out in site.pp and add the following line at the top:

```
hiera_include('classes')
```

- Add the following lines in common.yaml:

```
classes:  
- lamp_stack
```