

# Puppet CI

---

A way of doing Continuous Integration

# Agenda

- ❖ About me
- ❖ CI - say what?
- ❖ Step by step implementation
- ❖ Demo
- ❖ Tips & tricks
- ❖ Questions?

A close-up photograph of a person's hand, wearing a dark blue sleeve, adjusting a white slider on a professional audio mixing console. The hand is positioned in the lower-left quadrant of the frame. The mixing console features various faders, knobs, and buttons, with a row of red-tipped knobs visible on the right side. The background is heavily blurred, showing out-of-focus lights in shades of green, yellow, and red, suggesting a stage or concert environment. A semi-transparent dark grey horizontal band spans the middle of the image, serving as a background for the text.

About me



## About me

DevOps @ Broadbean Technology

Puppet user since October 2015

A newbie who is excited about new technologies

Any feedback: [jgruzewski@gmail.com](mailto:jgruzewski@gmail.com)



A close-up photograph of a person's hand, wearing a dark suit sleeve, adjusting a white slider on a professional audio mixing console. The hand is positioned in the lower-left quadrant of the frame. The mixing console has various faders and knobs, with some red and blue lights visible on the right side. The background is heavily blurred, showing out-of-focus bokeh lights in shades of green, yellow, and red, suggesting a stage or concert environment. A semi-transparent dark grey rectangular box is overlaid on the left side of the image, containing the text 'CI - say what?'. A thin vertical green line is positioned to the right of the text.

CI - say what?

# CI

## Continuous Integration

Continuous Integration is a software development practice where **members of a team integrate their work frequently**, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is **verified by an automated build** (including test) to detect integration errors as **quickly as possible**.

**I DON'T ALWAYS  
TEST MY CODE**

**BUT WHEN I DO, I  
DO IT IN PRODUCTION**

# Where CI can be placed?

Local

**Remote  
(central)**



A close-up photograph of a person's hand adjusting a white slider on a professional audio mixing console. The hand is positioned in the lower-left quadrant, with the index finger pressing down on the slider. The mixing console surface is metallic and features various other sliders and knobs, some of which are blurred in the background. The background is filled with out-of-focus, colorful bokeh lights in shades of blue, green, and red, suggesting a stage or studio environment. A semi-transparent dark grey rectangular box is overlaid on the left side of the image, containing the text "Step by step implementation" in white. A thin vertical green line is positioned to the right of the text.

Step by step  
implementation

# Puppet CI PIPELINE

Step 1

**Compilation**

- ❖ Puppet Parser
- ❖ Puppet Rspec
- ❖ Parallel Tests

Step 2

**Unit / Module**

- ❖ Beaker Rspec
- ❖ Puppet Rspec

Step 3

**End-to-End**

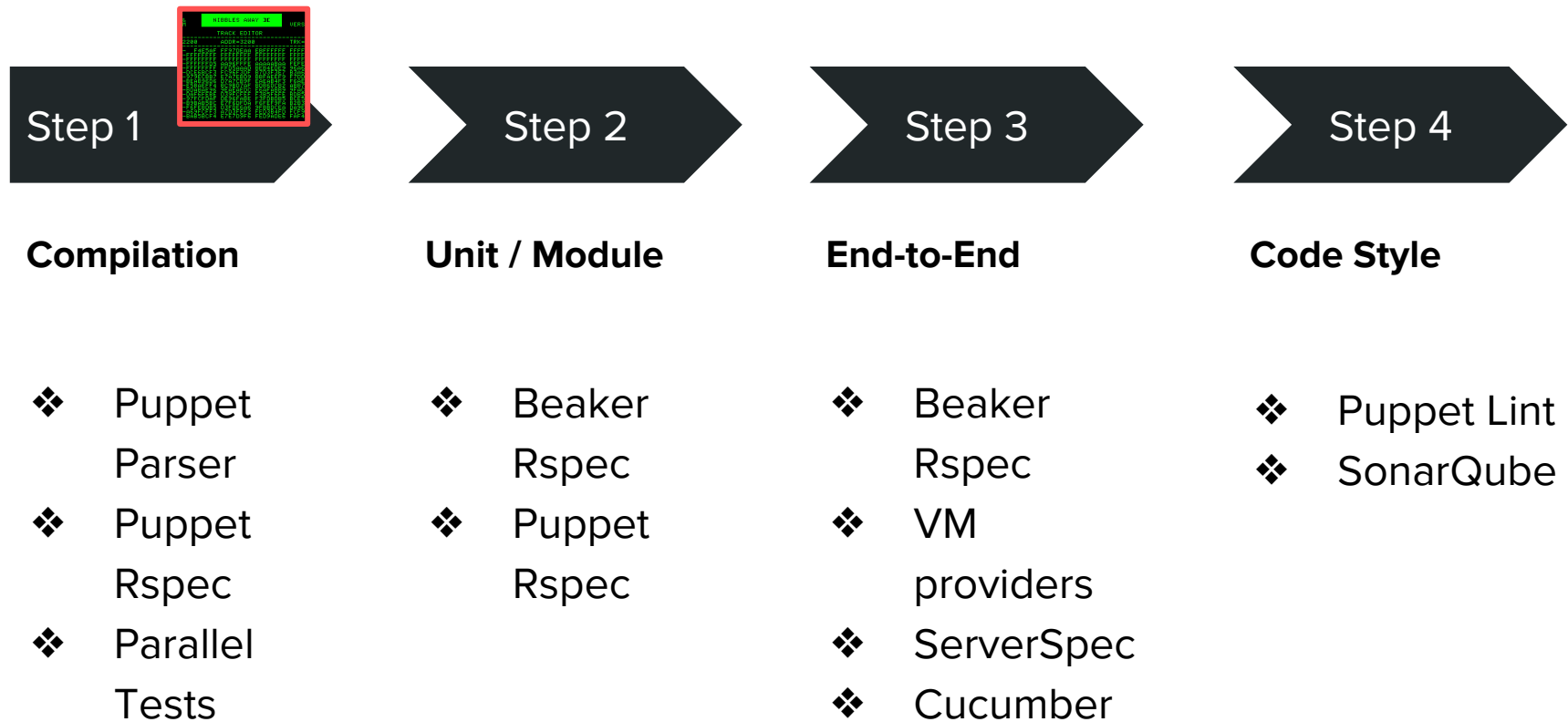
- ❖ Beaker Rspec
- ❖ VM providers
- ❖ ServerSpec
- ❖ Cucumber

Step 4

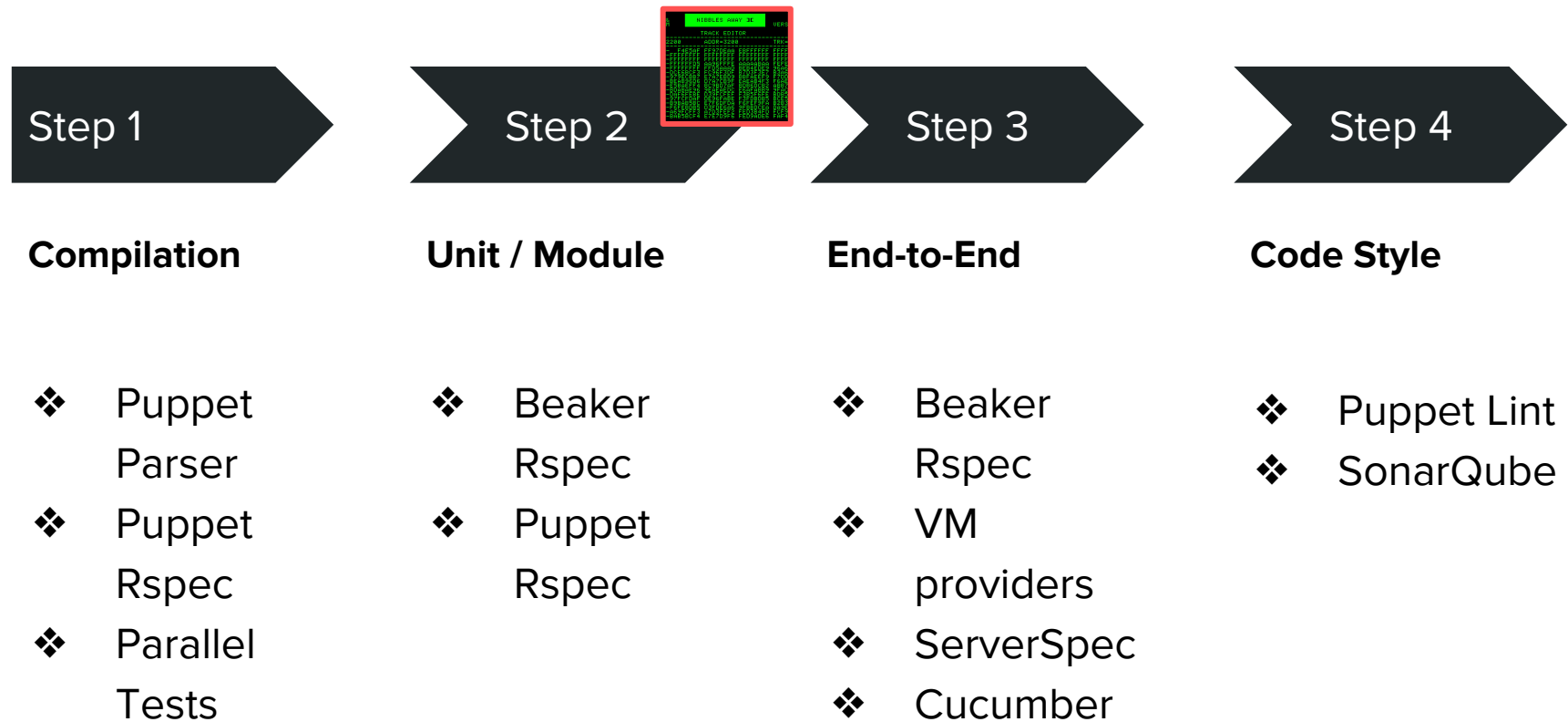
**Code Style**

- ❖ Puppet Lint
- ❖ SonarQube

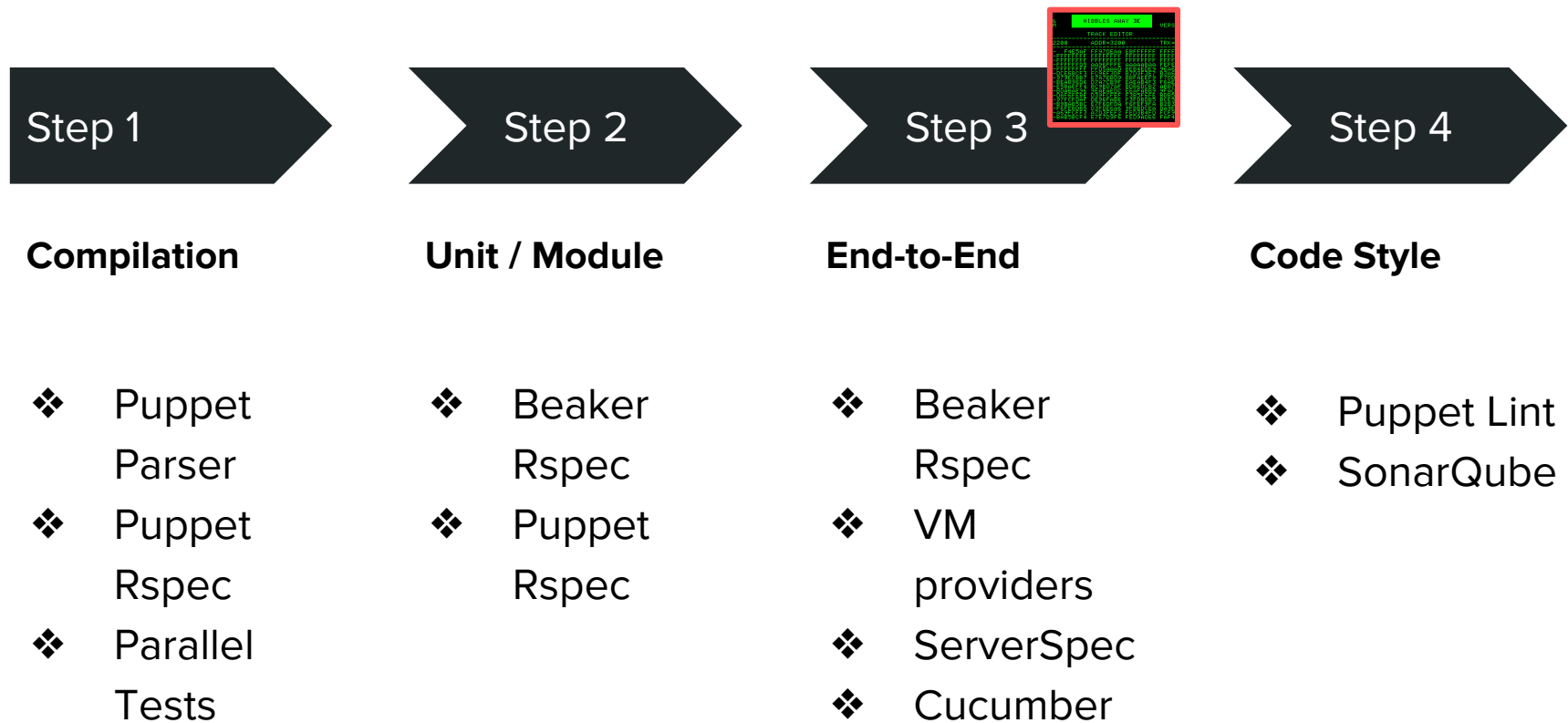
# Puppet CI PIPELINE



# Puppet CI PIPELINE



# Puppet CI PIPELINE



# Puppet CI PIPELINE

Step 1

**Compilation**

- ❖ Puppet Parser
- ❖ Puppet Rspec
- ❖ Parallel Tests

Step 2

**Unit / Module**

- ❖ Beaker Rspec
- ❖ Puppet Rspec

Step 3

**End-to-End**

- ❖ Beaker Rspec
- ❖ VM providers
- ❖ ServerSpec
- ❖ Cucumber

Step 4

**Code Style**

- ❖ Puppet Lint
- ❖ SonarQube





# Puppet CI PIPELINE

Step 1

**Compilation**

- ❖ Puppet Parser
- ❖ **Puppet Rspec**
- ❖ **Parallel Tests**

Step 2

**Unit / Module**

- ❖ Beaker Rspec
- ❖ Puppet Rspec

Step 3

**End-to-End**

- ❖ **Beaker Rspec**
- ❖ **VM providers**
- ❖ **ServerSpec**
- ❖ Cucumber

Step 4

**Code Style**

- ❖ Puppet Lint
- ❖ SonarQube

# Step 1 (Compilation)

## Requirements

- Puppet :)
- Puppet-Rspec
- Parallel Tests

## Assumptions

Runs for every role

# Step 1 (Compilation)

Example test:

```
require 'spec_helper'
```

```
let(:facts) { Facts.get('<role>') }
```

```
describe "<role>" do
```

```
  it { is_expected.to compile.with_all_deps }
```

```
end
```

# Step 1 (Compilation)

**Example test:**

```
require 'spec_helper'
```

```
let(:facts) { Facts.get('<role>') }
```

```
describe "<role>" do
```

```
  it { is_expected.to compile.with_all_deps }
```

```
end
```

# Step 1 (Compilation)

**Example test:**

```
require 'spec_helper'
```

```
let(:facts) { Facts.get('<role>') }
```

```
describe "<role>" do
```

```
  it { is_expected.to compile.with_all_deps }
```

```
end
```

# Step 1 (Compilation)

## spec\_helper.rb

```
require 'rubygems'  
require 'rspec-puppet'  
require 'facts'
```

```
puppet_path = '/etc/puppetlabs/code/environments/production'
```

```
RSpec.configure do |config|  
  config.module_path = File.join(puppet_path, 'modules')  
  config.hiera_config = File.join(puppet_path, 'hiera.yaml')  
  config.include Facts  
  config.expect_with :rspec do |c|  
    c.syntax = :expect  
  end  
end
```



# Step 1 (Compilation)

**Example test:**

```
require 'spec_helper'
```

```
let(:facts) { Facts.get('<role>') }
```

```
describe "<role>" do
```

```
  it { is_expected.to compile.with_all_deps }
```

```
end
```

# Step 1 (Compilation)

## facts.rb

```
module Facts
  def self.get(role, facts={})
    default_facts = {
      :role           => role,
      :fqdn           => role,
      :hostname       => role,
      :memorysize_mb  => 4098,
      :operatingsystem => 'Ubuntu',
      :osfamily       => 'Debian',
      :lsbdistcodename => 'trusty',
      :ipaddress       => '192.168.0.1',
    }

    default_facts.update(facts)
  end
end
```

# Step 1 (Compilation)

Example test:

```
require 'spec_helper'
```

```
let(:facts) { Facts.get('<role>') }
```

```
describe "<role>" do
```

```
  it { is_expected.to compile.with_all_deps }
```

```
end
```

# Step 1 (Compilation)

## Example output:

Failures:

- 1) **<role>** should compile into a catalogue without dependency cycles

Failure/Error: it { is\_expected.to compile.with\_all\_deps }

error during compilation: Could not parse for environment rp\_env:

Syntax error at 'prefix\_name'; expected '}' at

/etc/puppet/manifests/nodes.pp:10 on node **<hostname>**

# ./spec/hosts/compile/**<role>**\_spec.rb:8:in `block (2 levels) in

<top (required)>'

## Step 3 (End-To-End)

### Requirements

- Beaker-Rspec
- Puppet-Rspec
- Vagrant
- Virtual Box

### Assumptions

Should run only for  
modified roles

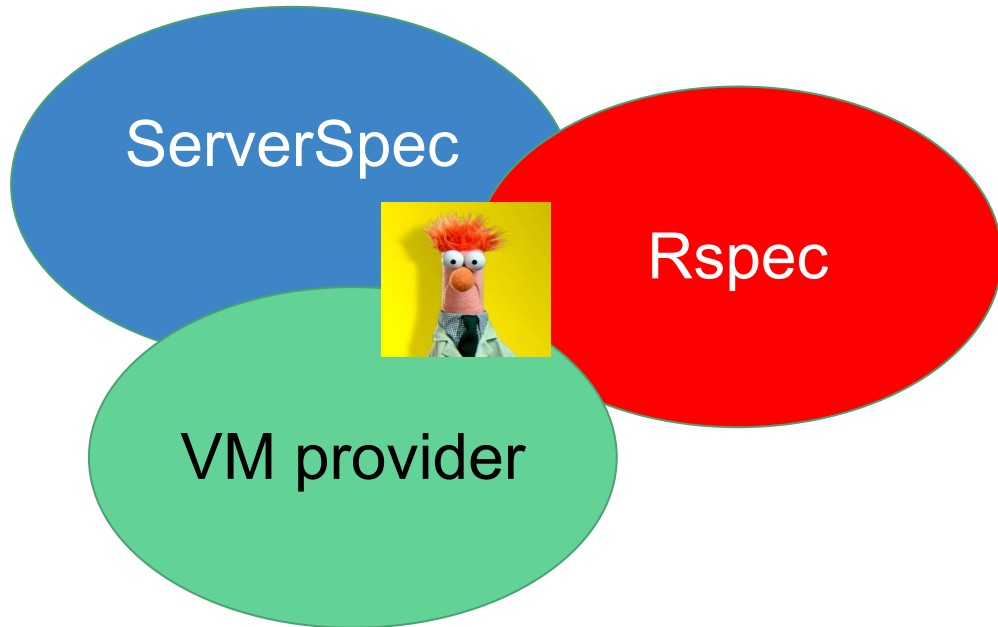
## Step 3 (End-To-End)

**Most Wanted!**



**MR BEAKER RSPEC**

Puppet Acceptance Testing Harness





# Step 3 (End-To-End)

## VM Providers:

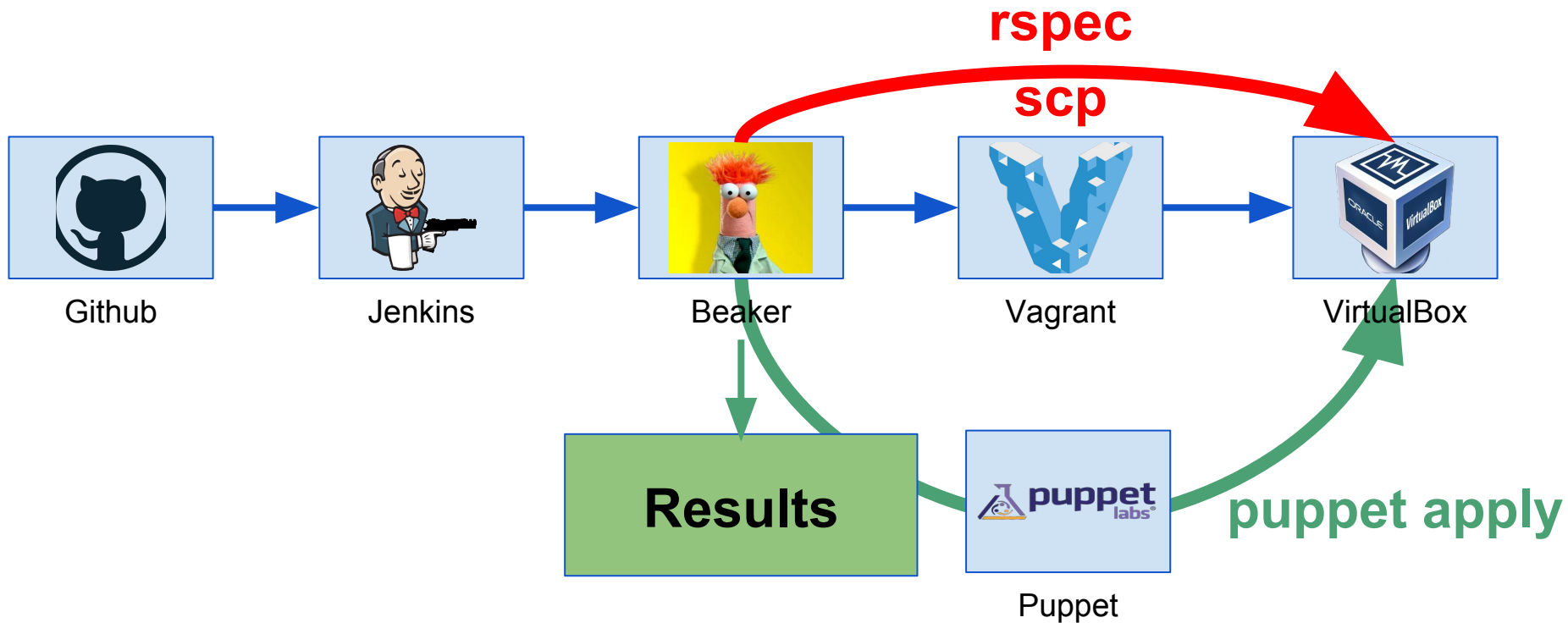
- ❖ AWS EC2
- ❖ Docker
- ❖ Google Compute Engine
- ❖ OpenStack
- ❖ Vagrant
  - Libvirt
  - Oracle VirtualBox
  - VMware Workstation
- ❖ VMware Fusion
- ❖ VMware vSphere

# Step 3 (End-To-End)

## VM Providers:

- ❖ AWS EC2
- ❖ Docker
- ❖ Google Compute Engine
- ❖ OpenStack
- ❖ **Vagrant**
  - Libvirt
  - **Oracle VirtualBox**
  - VMware Workstation
- ❖ VMware Fusion
- ❖ VMware vSphere

# Step 3 (End-To-End)



# Step 3 (End-To-End)

## Example test:

```
require 'spec_helper_acceptance'
```

```
describe 'wordpress' do
```

```
  it 'should run successfully' do
```

```
    agent = only_host_with_role(hosts, 'agent')
```

```
    apply_puppet = ". /etc/profile && puppet apply /etc/puppetlabs/code/environments/production/manifests/site.pp"
```

```
    on(agent, apply_puppet) { assert_equal(0, exit_code) }
```

```
  end
```

```
describe package('mysql-server') do
```

```
  it { should be_installed }
```

```
end
```

```
describe port(80) do
```

```
  it { should be_listening }
```

```
end
```

```
end
```

# Step 3 (End-To-End)

**spec\_helper\_acceptance.rb:**

```
require 'beaker-rspec'
```

```
unless ENV['RS_PROVISION'] == 'no'  
  hosts.each do |host|
```

```
    if host['roles'].include?('agent')
```

```
      agent = host
```

```
      files = ['manifests', 'modules']
```

```
      on agent, "wget https://apt.puppetlabs.com/puppetlabs-release-pc1-precise.deb"
```

```
      on agent, "sudo dpkg -i puppetlabs-release-pc1-precise.deb"
```

```
      on agent, "apt-get update -y"
```

```
      on agent, "apt-get install -y puppet-agent"
```

```
      files.each do |file|
```

```
        scp_to agent, File.expand_path(file, '/etc/puppetlabs/'), "/etc/puppetlabs/"
```

```
      end
```

```
    end
```

```
  end
```

```
end
```

# Step 3 (End-To-End)

```
RSpec.configure do |c|  
  # Project root  
  proj_root = File.expand_path(File.join(File.dirname(__FILE__), '..'))  
  
  # Readable test descriptions  
  c.formatter = :documentation  
  
  c.before :suite do  
    hosts.each do |host|  
      shell(''. /etc/profile', { :acceptable_exit_codes => [0,1] })  
    end  
  end  
end
```



# Step 3 (End-To-End)

## default.yml:

### HOSTS:

ubuntu-agent:

roles:

- default
- agent

platform: ubuntu-12.04-i386

box: base

hypervisor: vagrant

vagrant\_memsize: 1024

ip: '192.168.57.130'

### CONFIG:

log\_level: debug

type: foss

# Step 3 (End-To-End)

## Example output:

Port "80"

```
ubuntu-agent 15:59:01$ /bin/sh -c netstat\ -tun\ |\ grep\ --\ :80\|
```

```
tcp      0      0 0.0.0.0:80      0.0.0.0:*      LISTEN
```

ubuntu-agent executed in **0.52 seconds**

should be listening

Destroying vagrant boxes

==> ubuntu-agent: Forcing shutdown of VM...

==> ubuntu-agent: Destroying VM and associated drives...

Finished in **3 minutes 32.5 seconds** (files took **3 minutes 4.4 seconds** to load)

**5 examples, 0 failures**

A close-up photograph of a person's hand, wearing a dark suit sleeve, adjusting a white slider on a professional audio mixing console. The hand is positioned in the lower-left quadrant of the frame. The mixing console features multiple tracks with faders and various knobs. In the background, out of focus, are several bright, circular bokeh lights in shades of green and blue, suggesting a stage or studio environment. The word "Demo" is overlaid in white text on the left side of the image.

Demo

# Tips and Tricks

A close-up photograph of a person's hand, wearing a dark blue sleeve, adjusting a white fader on a professional audio mixing console. The hand is positioned in the lower-left quadrant of the frame. The mixing console features multiple tracks with faders and various knobs, extending towards the right. The background is heavily blurred, showing out-of-focus lights in shades of green, yellow, and red, suggesting a stage or concert environment. A semi-transparent dark grey horizontal band spans the middle of the image, containing the text 'Tips and Tricks' in white. A thin, vertical green line is positioned to the right of the text, extending from the top of the dark band to the bottom of the image.



# Tips

- ❖ Try with pre-build modules first
- ❖ Beaker has local environment “BEAKER\_destroy” (no, onpass)
- ❖ LXC could be your friend (vagrant-lxc plugin)
- ❖ Use base image with pre-baked software to speed up your tests
- ❖ Configure everything by hand and then automate it
- ❖ Jenkins has plugin to comment your Commits / Pull Request

# Tips

quick fix to save the chicken or the egg scenario

Overview

Commits

Activity

Author



Reviewers



Description

*No description*



1 of 1 passed




Stop watching



Learn about pull requests

# Tips

## Builds

 puppet-1stage

test-puppet-1st-stage: be5f29d294f7 into master

2 days ago

Close

# Tips

## Comments (5)



**jenkins broadbean**

[*BuildFinished* **puppet-test-1st-stage**] [a265f31e3fba](#) into [e0cca199a4ad](#)

**SUCCESS** - [http://\[redacted\]/job/puppet-test-1st-stage/74/](http://[redacted]/job/puppet-test-1st-stage/74/)

Reply • Delete • Create task • 2015-11-09



# Handy links

- ❖ <https://github.com/puppetlabs/beaker/blob/master/docs/README.md>
- ❖ <http://www.xkyle.com/what-happens-when-you-run-puppet-tests/>
- ❖ <https://puppetlabs.com/presentations/workshop-know-you-push-go-using-beaker-acceptance-test-framework-test-your-puppet>
- ❖ <http://betterspecs.org/>
- ❖ <https://puppetlabs.com/blog/the-next-generation-of-puppet-module-testing>

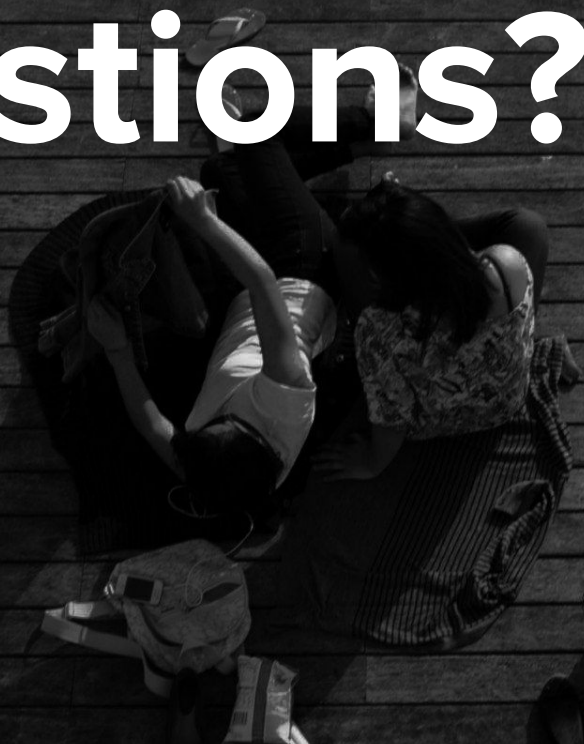
# Handy links

- ❖ <https://github.com/puppetlabs/beaker/blob/master/docs/README.md>
- ❖ <http://www.xkyle.com/what-happens-when-you-run-puppet-tests/>
- ❖ <https://puppetlabs.com/presentations/workshop-know-you-push-go-using-beaker-acceptance-test-framework-test-your-puppet>
- ❖ <http://betterspecs.org/>
- ❖ <https://puppetlabs.com/blog/the-next-generation-of-puppet-module-testing>

And problems start here ...

.....

# Questions?



**THEY SAID I COULD BE ANYTHING I  
WANTED...**



**SO I BECAME A GOD**

quickmeme.com

# Thank you!

:)

Sometimes I question  
my sanity,  
but the unicorn  
and gummy bears  
told me  
I'm fine!

