# beaker-rspec and you: module testing for a brighter future

Alice Nodelman, QE

things break

# Alice Nodelman

alice@puppetlabs.com

anode (freenode #puppet-dev)

alicenode (twitter)

anodelman (github)

# acceptance testing

we make promises, we test to keep them

# what’s beaker?

# we set things on fire,
# so that users don’t get burnt

# the formula

Ruby

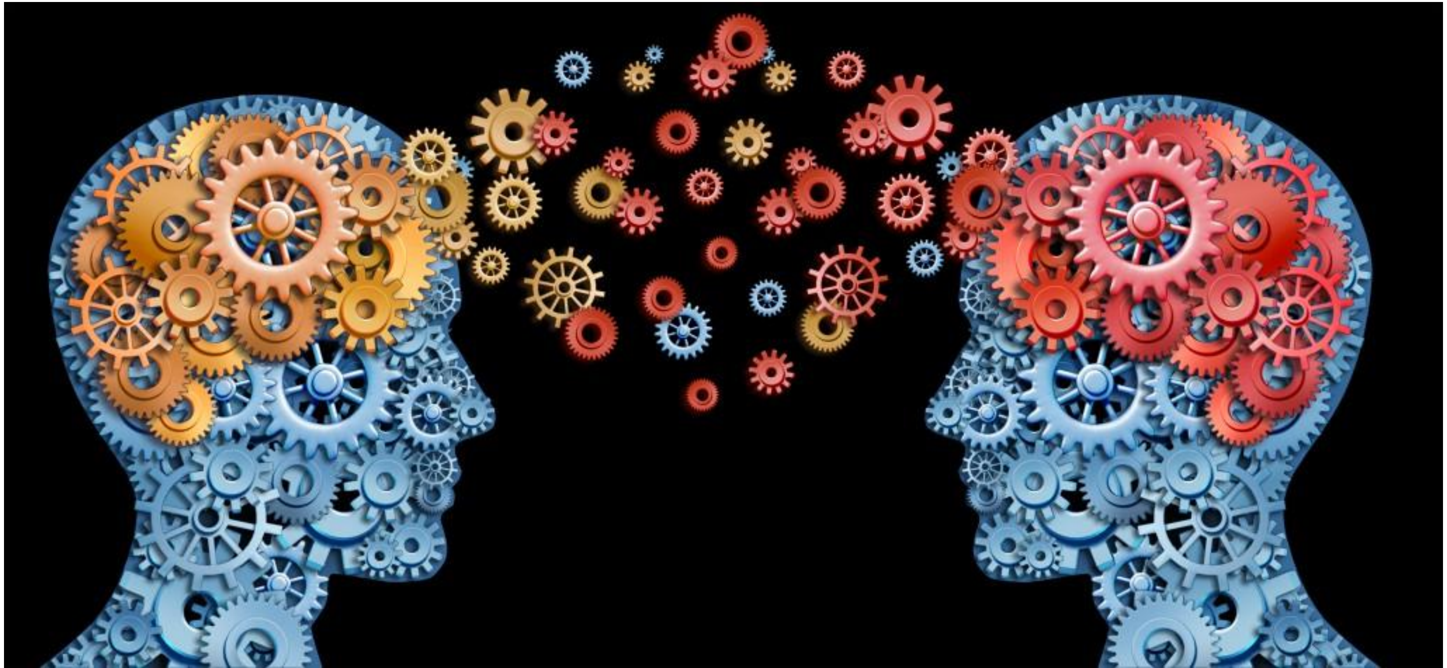
+ DSL

+ virtualized SUTs

= success

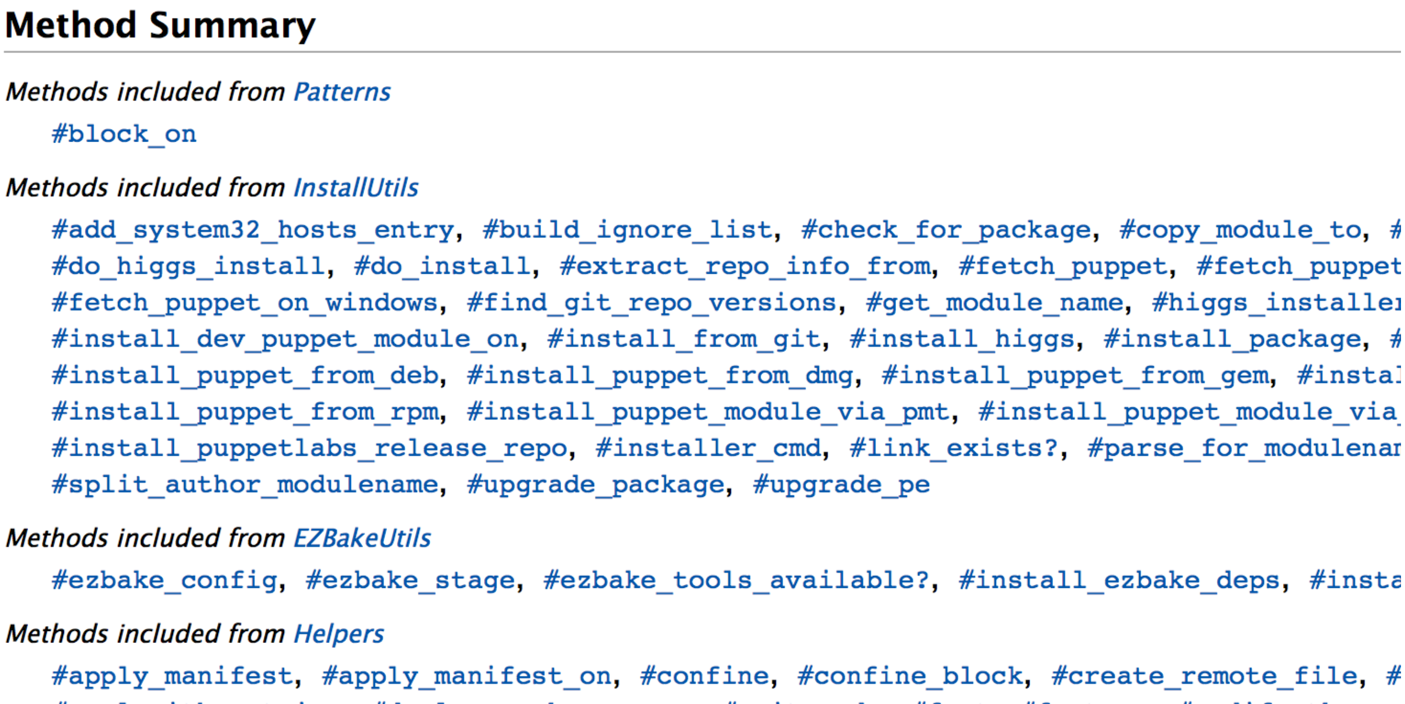
# why ruby?

## we know it and so do other people



# the DSL

rubydoc.info



## Method Summary

*Methods included from Patterns*

#block_on

*Methods included from InstallUtils*

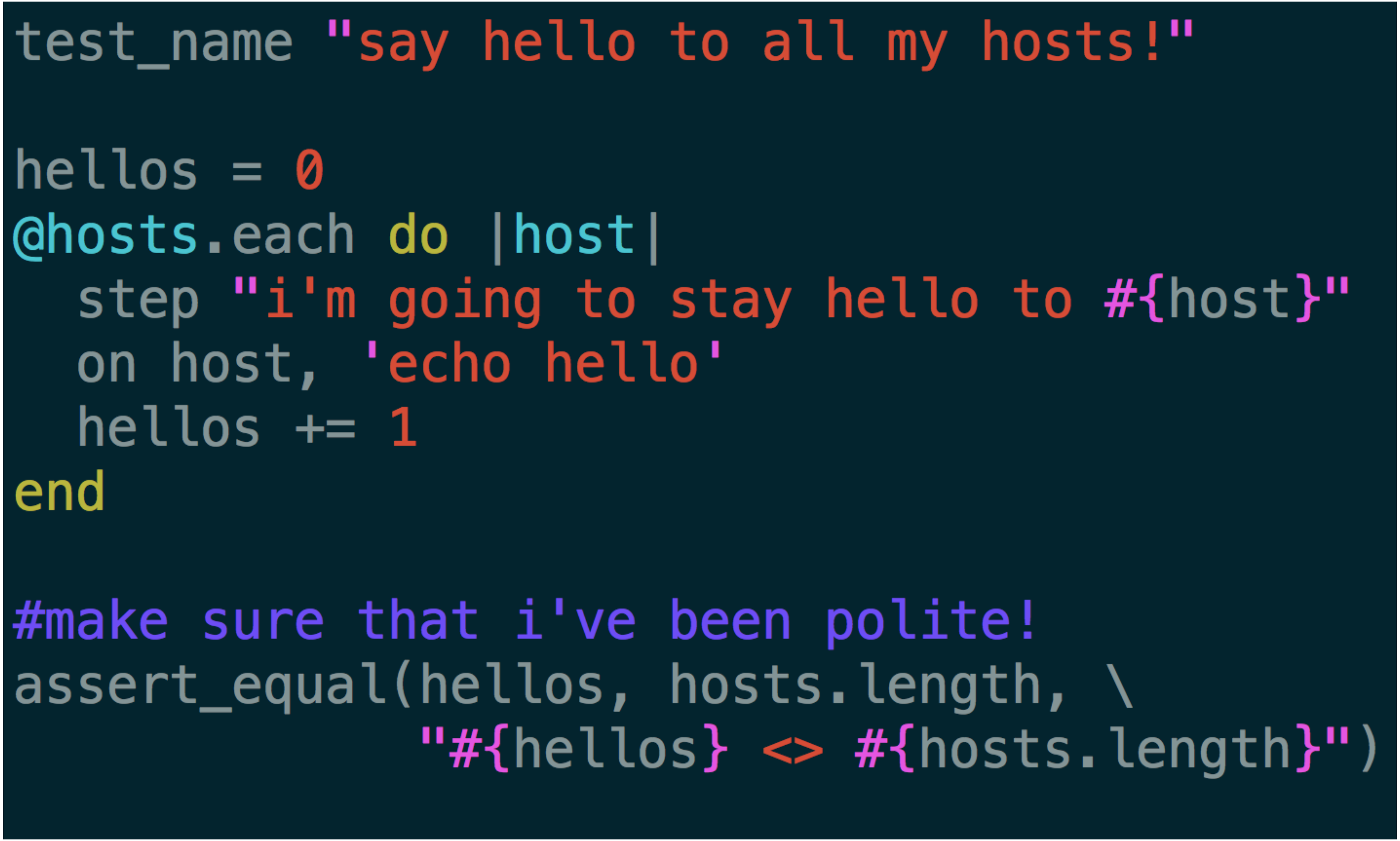
#add_system32_hosts_entry, #build_ignore_list, #check_for_package, #copy_module_to, #
#do_higgs_install, #do_install, #extract_repo_info_from, #fetch_puppet, #fetch_puppet
#fetch_puppet_on_windows, #find_git_repo_versions, #get_module_name, #higgs_installe
#install_dev_puppet_module_on, #install_from_git, #install_higgs, #install_package, #
#install_puppet_from_deb, #install_puppet_from_dmg, #install_puppet_from_gem, #instal
#install_puppet_from_rpm, #install_puppet_module_via_pmt, #install_puppet_module_via
#install_puppetlabs_release_repo, #installer_cmd, #link_exists?, #parse_for_modulenar
#split_author_modulename, #upgrade_package, #upgrade_pe

*Methods included from EZBakeUtils*

#ezbake_config, #ezbake_stage, #ezbake_tools_available?, #install_ezbake_deps, #insta

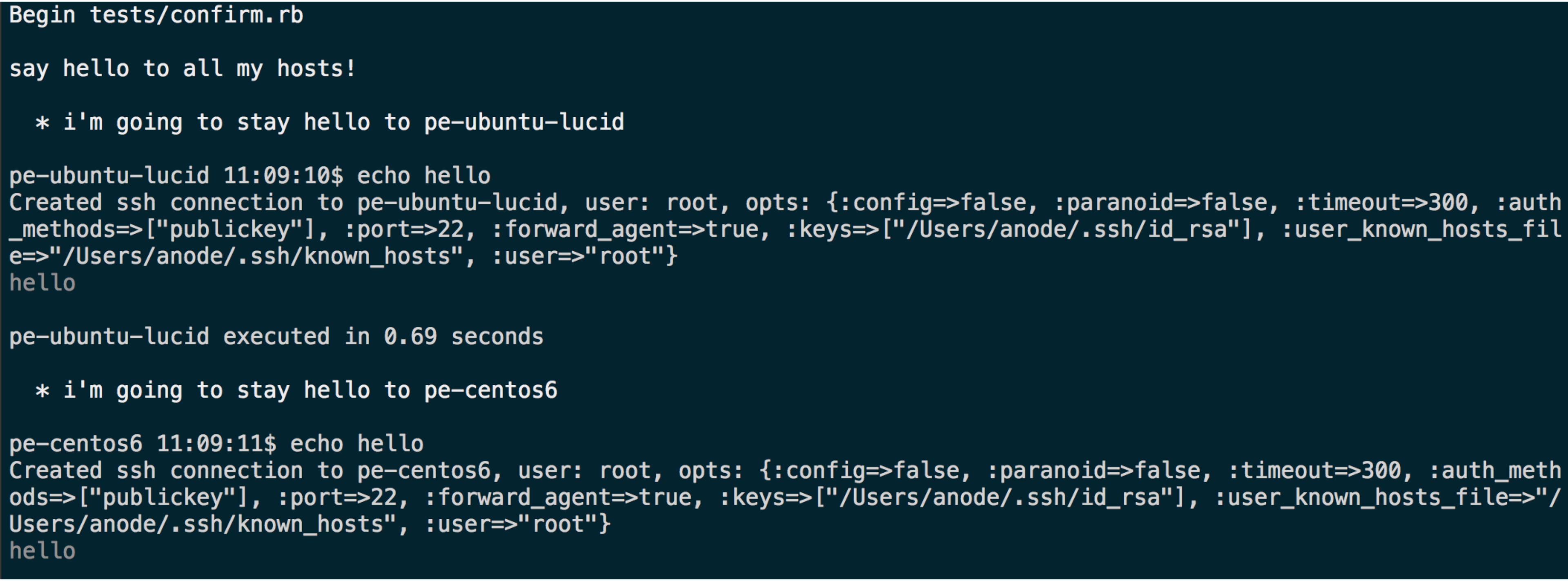
*Methods included from Helpers*

#apply_manifest, #apply_manifest_on, #confine, #confine_block, #create_remote_file, #



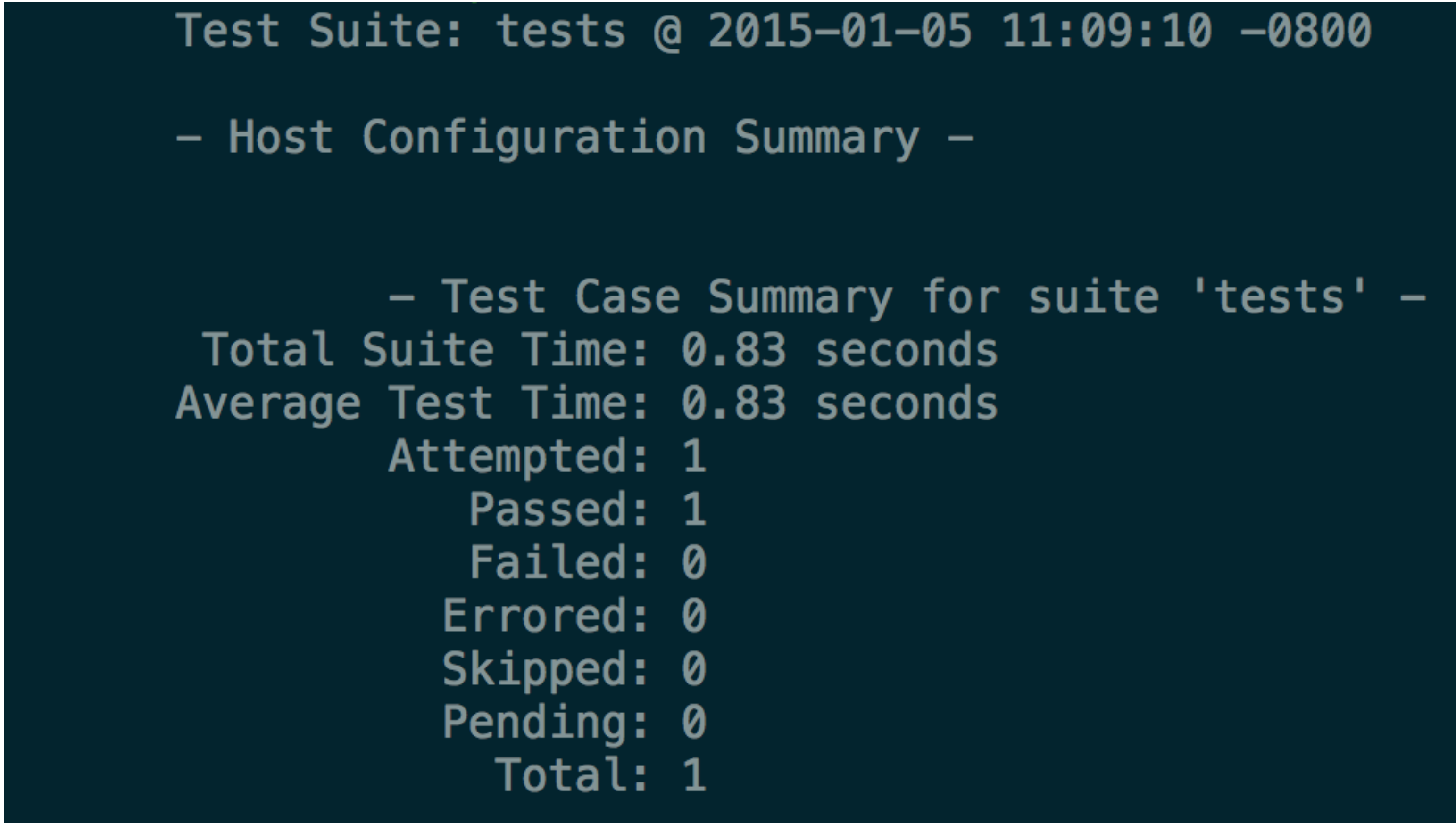
```
test_name "say hello to all my hosts!"

hellos = 0
@hosts.each do |host|
  step "i'm going to stay hello to #{host}"
  on host, 'echo hello'
  hellos += 1
end

#make sure that i've been polite!
assert_equal(hellos, hosts.length, \
             "#{hellos} <> #{hosts.length}")
```

# what you see in the console



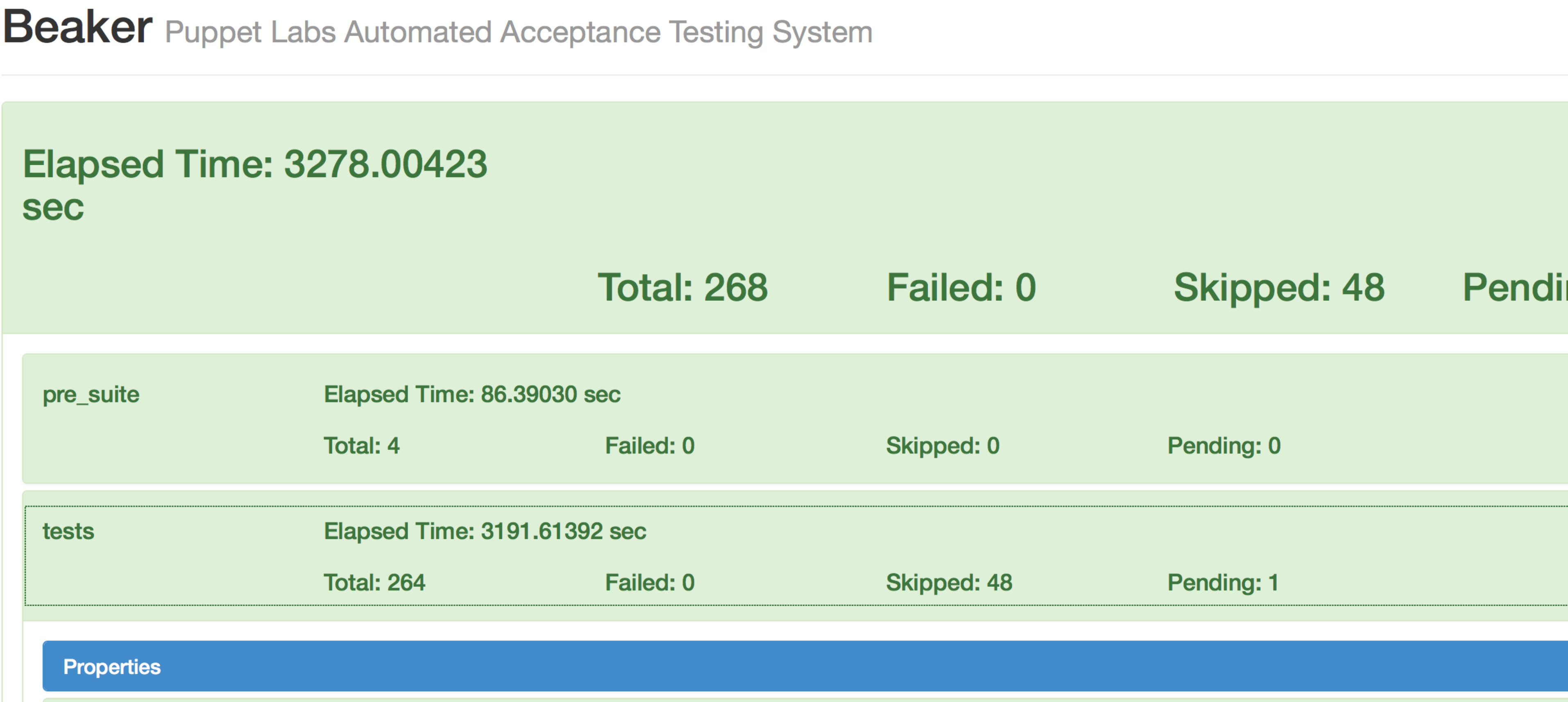
```
Begin tests/confirm.rb

say hello to all my hosts!

  * i'm going to stay hello to pe-ubuntu-lucid

pe-ubuntu-lucid 11:09:10$ echo hello
Created ssh connection to pe-ubuntu-lucid, user: root, opts: {:config=>false, :paranoid=>false, :timeout=>300, :auth
_methods=>["publickey"], :port=>22, :forward_agent=>true, :keys=>["/Users/anode/.ssh/id_rsa"], :user_known_hosts_fil
e=>"/Users/anode/.ssh/known_hosts", :user=>"root"}
hello

pe-ubuntu-lucid executed in 0.69 seconds

  * i'm going to stay hello to pe-centos6

pe-centos6 11:09:11$ echo hello
Created ssh connection to pe-centos6, user: root, opts: {:config=>false, :paranoid=>false, :timeout=>300, :auth_meth
ods=>["publickey"], :port=>22, :forward_agent=>true, :keys=>["/Users/anode/.ssh/id_rsa"], :user_known_hosts_file=>"/
Users/anode/.ssh/known_hosts", :user=>"root"}
hello
```

# console summary



```
Test Suite: tests @ 2015-01-05 11:09:10 -0800

- Host Configuration Summary -


        - Test Case Summary for suite 'tests' -
 Total Suite Time: 0.83 seconds
Average Test Time: 0.83 seconds
        Attempted: 1
           Passed: 1
           Failed: 0
          Errored: 0
          Skipped: 0
          Pending: 0
            Total: 1
```

# Beaker Junit Output - so pretty!



Beaker Puppet Labs Automated Acceptance Testing System

Elapsed Time: 3278.00423 sec

Total: 268 Failed: 0 Skipped: 48 Pendi

pre_suite Elapsed Time: 86.39030 sec

Total: 4 Failed: 0 Skipped: 0 Pending: 0

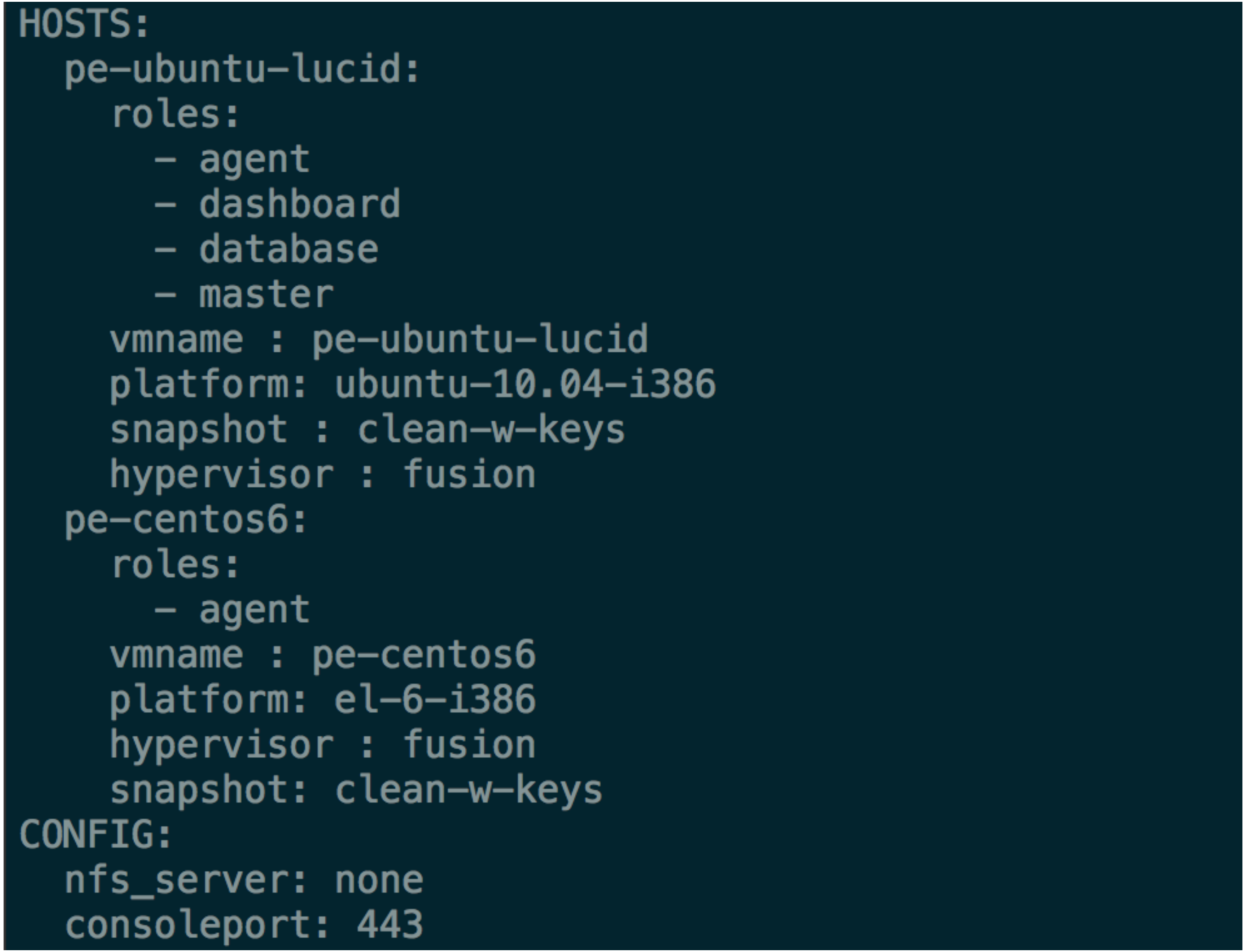
tests Elapsed Time: 3191.61392 sec

Total: 264 Failed: 0 Skipped: 48 Pending: 1

Properties

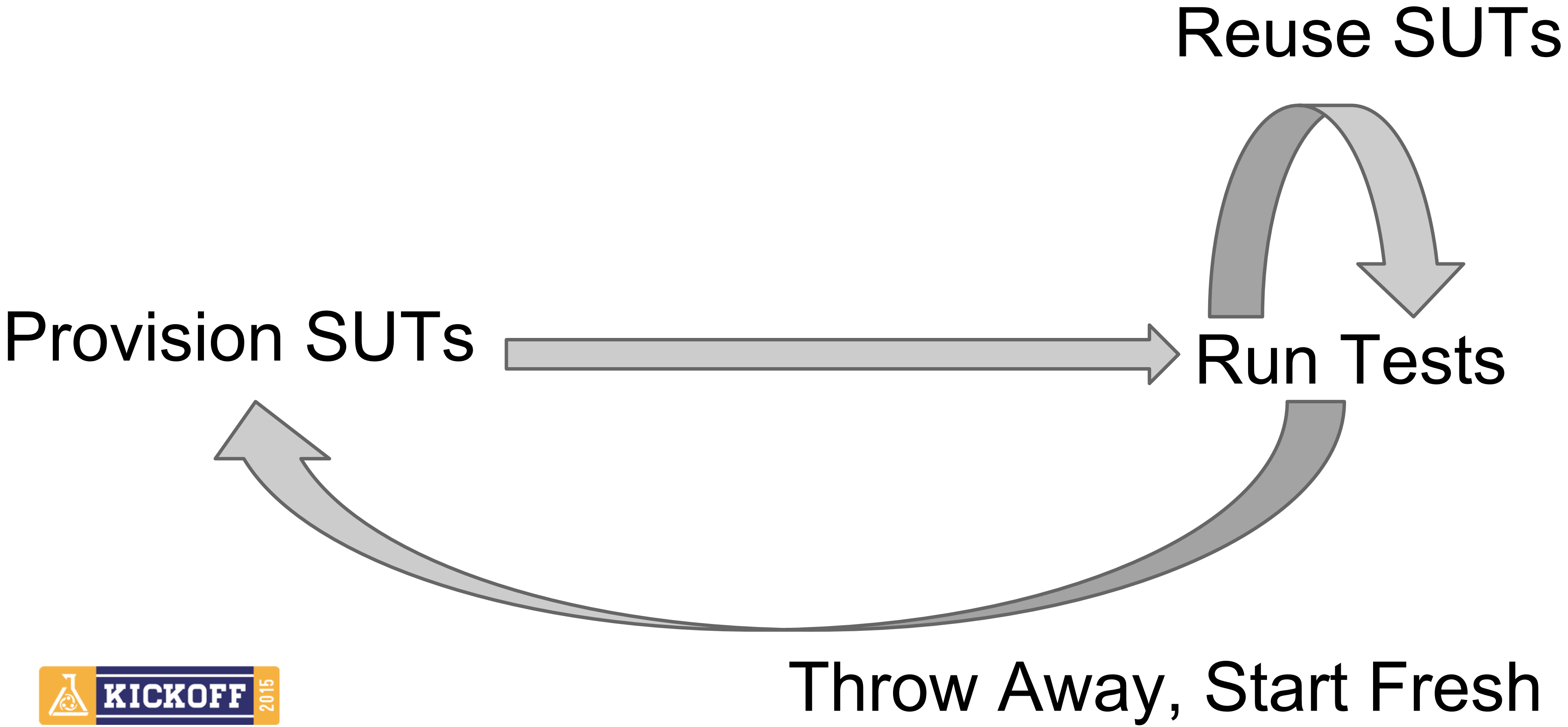
# virtualization

- vSphere
- fusion
- vagrant
- openstack
- docker
- AWS
- GCE
- aix + solaris



```
HOSTS:
  pe-ubuntu-lucid:
    roles:
      - agent
      - dashboard
      - database
      - master
    vmname : pe-ubuntu-lucid
    platform: ubuntu-10.04-i386
    snapshot : clean-w-keys
    hypervisor : fusion
  pe-centos6:
    roles:
      - agent
    vmname : pe-centos6
    platform: el-6-i386
    hypervisor : fusion
    snapshot: clean-w-keys
CONFIG:
  nfs_server: none
  consoleport: 443
```

# the workflow



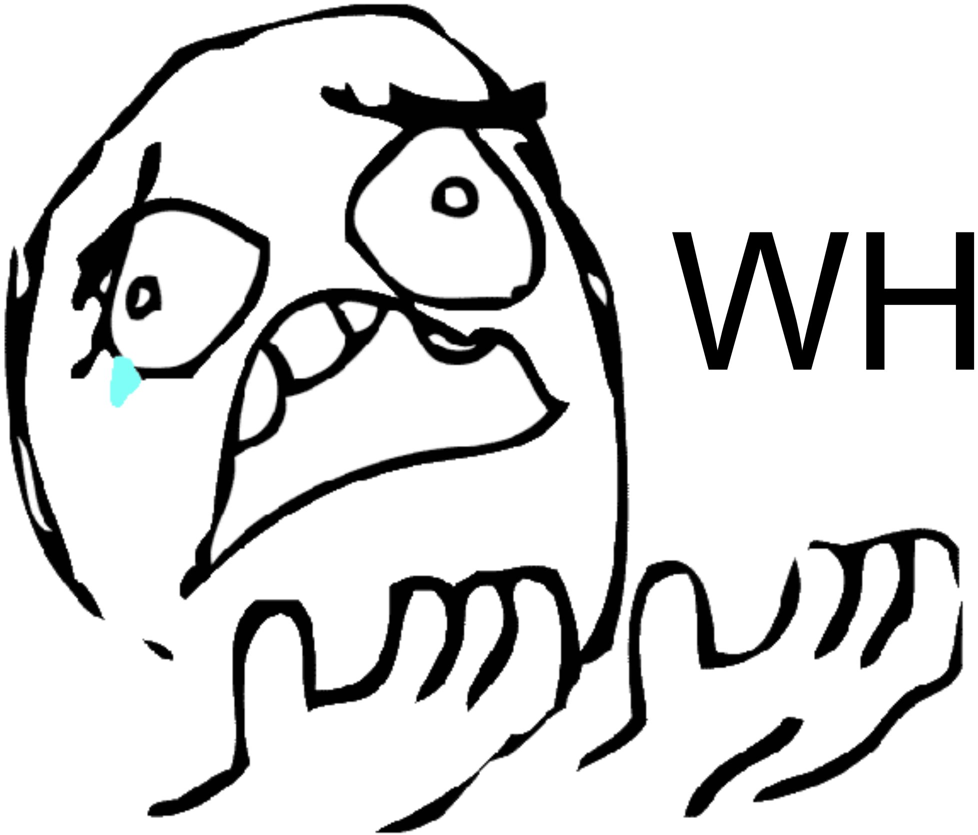
Reuse SUTs

Provision SUTs

Run Tests

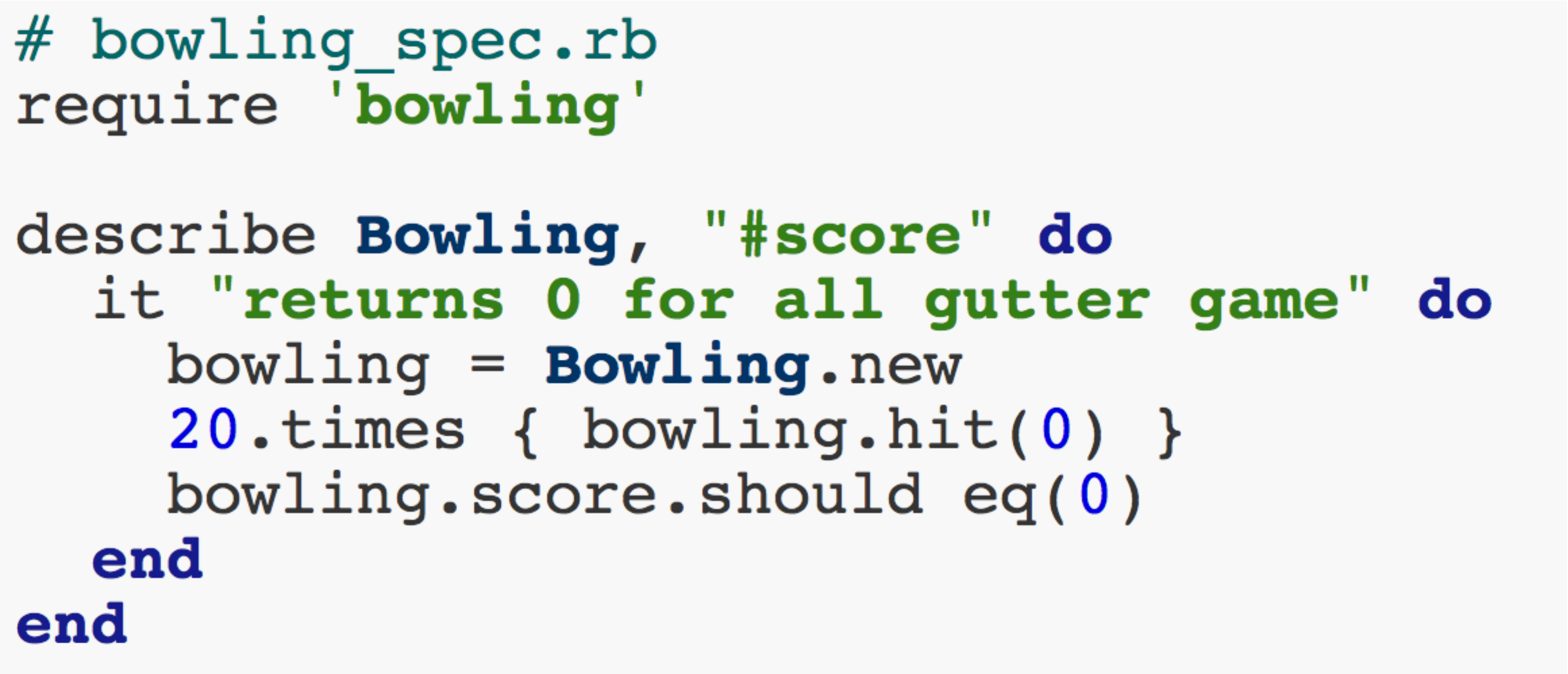
Throw Away, Start Fresh

# so, beaker-rspec?



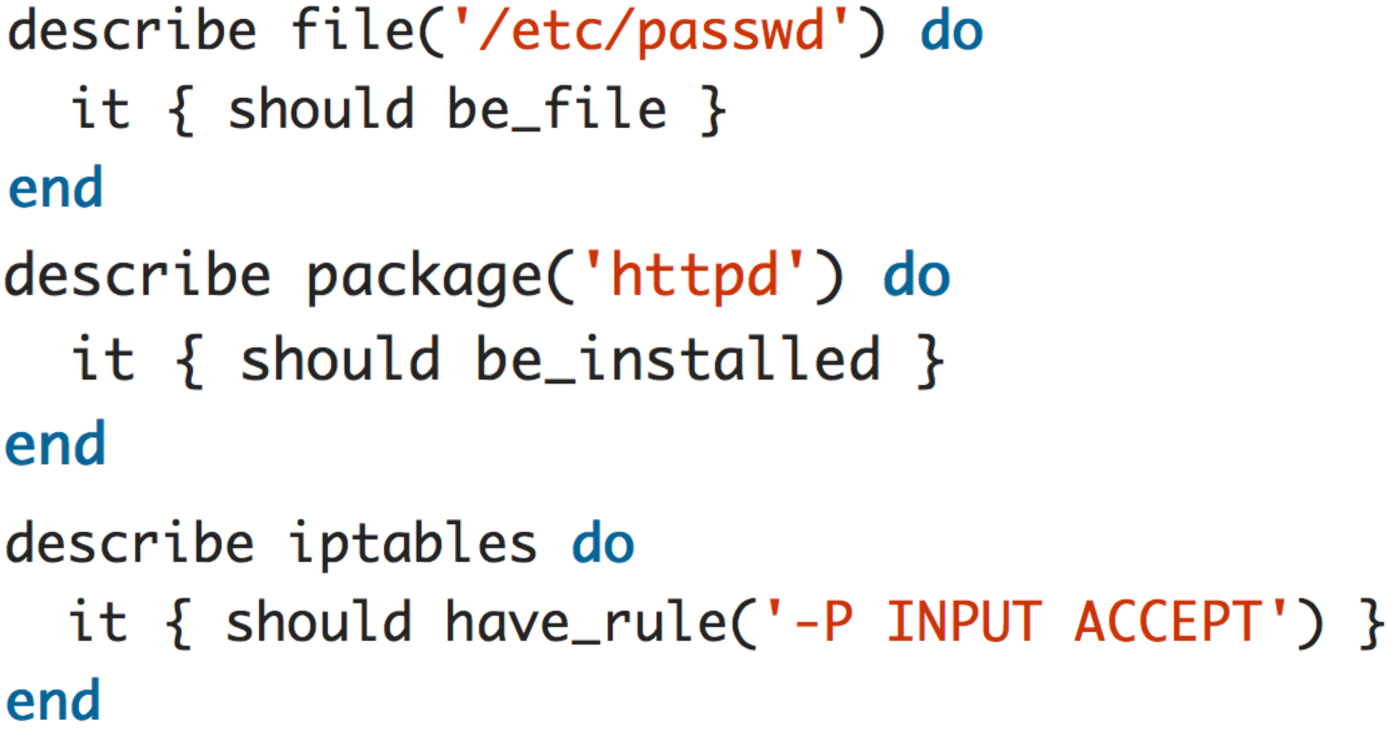
WHY?

# RSpec



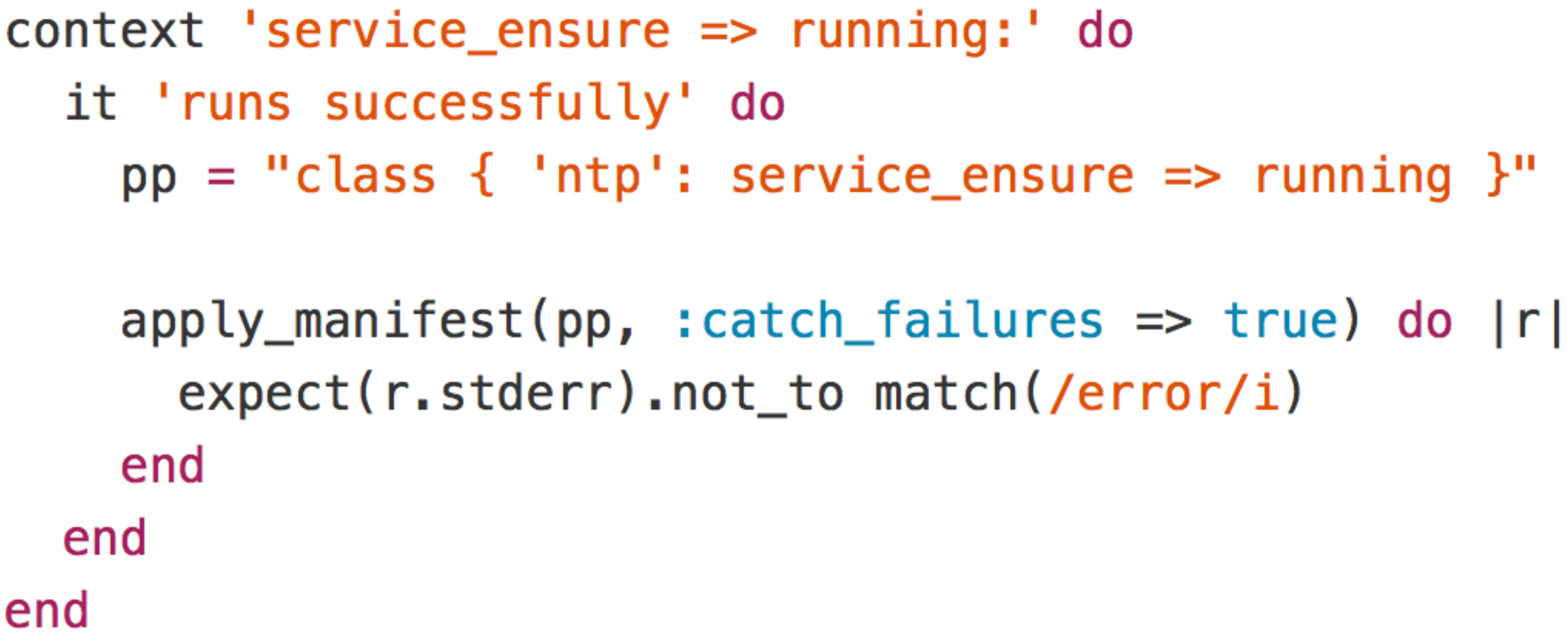
```ruby
# bowling_spec.rb
require 'bowling'

describe Bowling, "#score" do
  it "returns 0 for all gutter game" do
    bowling = Bowling.new
    20.times { bowling.hit(0) }
    bowling.score.should eq(0)
  end
end
```

# Serverspec



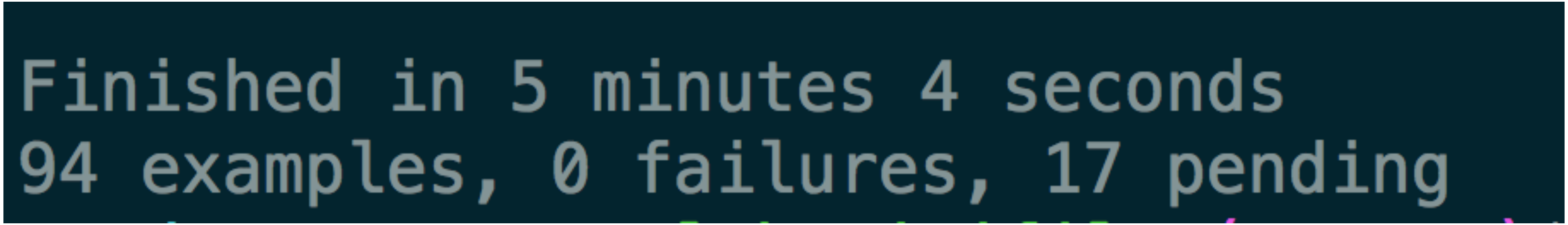
```
describe file('/etc/passwd') do
  it { should be_file }
end
describe package('httpd') do
  it { should be_installed }
end
describe iptables do
  it { should have_rule('-P INPUT ACCEPT') }
end
```

# a beaker-rspec test



```
context 'service_ensure => running:' do
  it 'runs successfully' do
    pp = "class { 'ntp': service_ensure => running }"

    apply_manifest(pp, :catch_failures => true) do |r|
      expect(r.stderr).not_to match(/error/i)
    end
  end
end
```

# console output



```
Finished in 5 minutes 4 seconds
94 examples, 0 failures, 17 pending
```

# Workshop!

# Run Tests Included in a Puppet Supported Module

RUN RUN RUN RUN RUN RUN

# part 1

## Steps

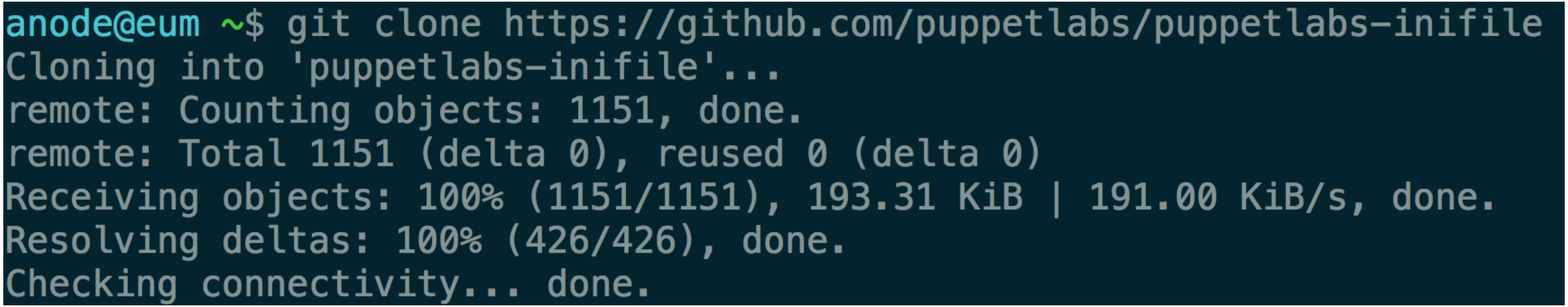
- install testing dependencies
- clone puppetlabs-inifile module
- set test parameters
- execute the puppetlabs-inifile beaker-rspec tests

# dependencies

- git
- Ruby 1.9+
- RubyGems
- Bundler
- Vagrant 1.5.2
- VirtualBox

# clone puppetlabs-inifile

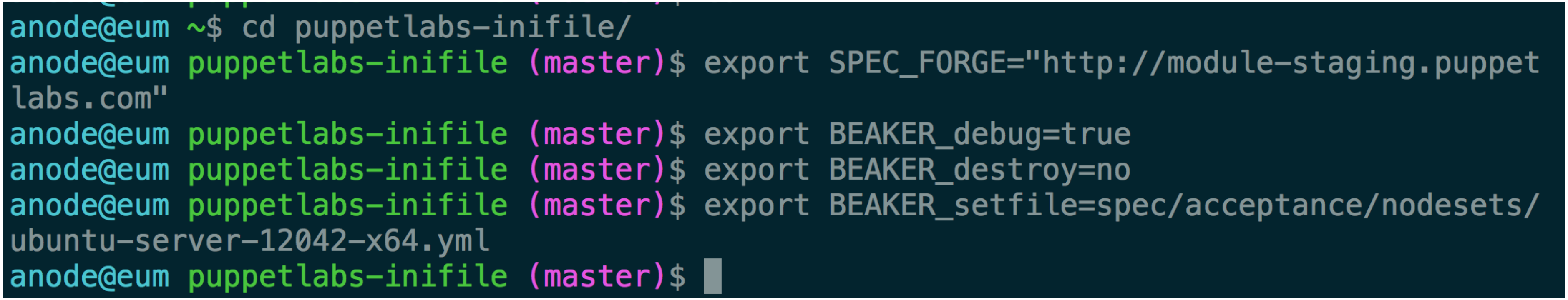
```
$ git clone https://github.com/puppetlabs/puppetlabs-inifile
```



```
anode@eum ~$ git clone https://github.com/puppetlabs/puppetlabs-inifile
Cloning into 'puppetlabs-inifile'...
remote: Counting objects: 1151, done.
remote: Total 1151 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (1151/1151), 193.31 KiB | 191.00 KiB/s, done.
Resolving deltas: 100% (426/426), done.
Checking connectivity... done.
```

# setup your test parameters

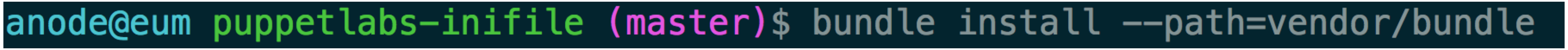
```
$ cd puppetlabs-inifile/
$ export SPEC_FORGE="http://module-staging.puppetlabs.com"
$ export BEAKER_debug=true
$ export BEAKER_destroy=no
$ export BEAKER_setfile=spec/acceptance/nodesets/ubuntu-server-12042-x64.yml
```



```
anode@eum ~$ cd puppetlabs-inifile/
anode@eum puppetlabs-inifile (master)$ export SPEC_FORGE="http://module-staging.puppet
labs.com"
anode@eum puppetlabs-inifile (master)$ export BEAKER_debug=true
anode@eum puppetlabs-inifile (master)$ export BEAKER_destroy=no
anode@eum puppetlabs-inifile (master)$ export BEAKER_setfile=spec/acceptance/nodesets/
ubuntu-server-12042-x64.yml
anode@eum puppetlabs-inifile (master)$ 
```

# install!

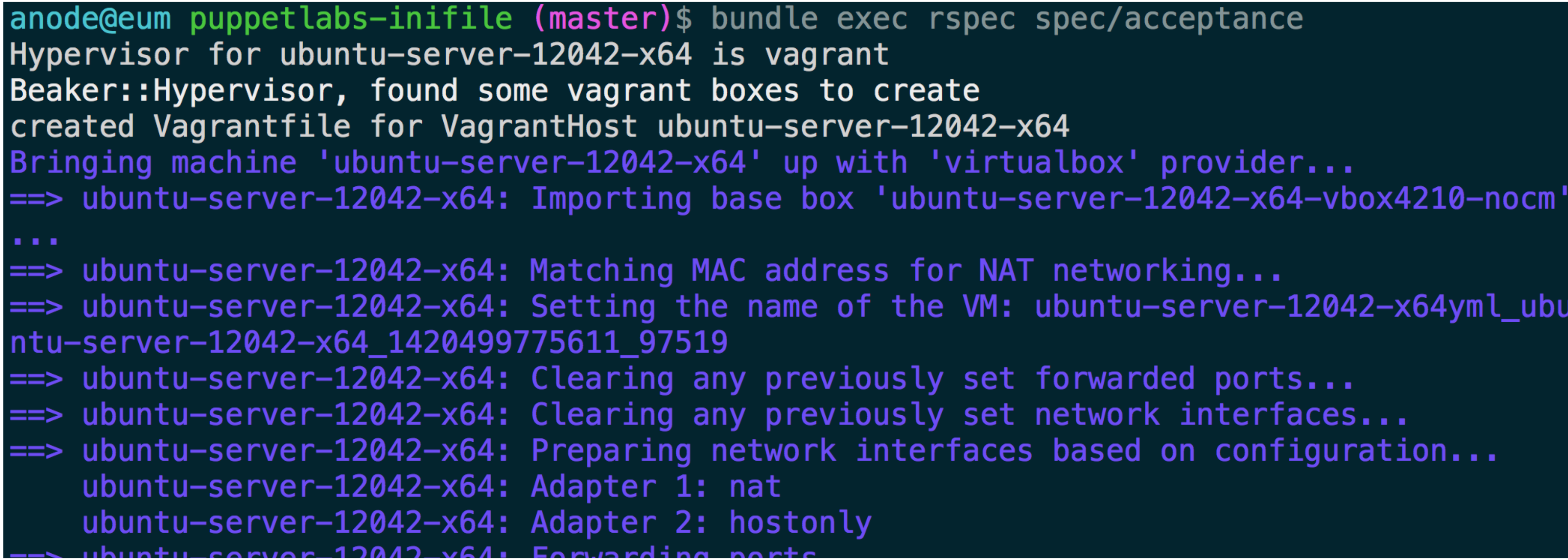
```
$ bundle install --path=vendor/bundle
```



```
anode@eum puppetlabs-inifile (master)$ bundle install --path=vendor/bundle
```

# execute the tests

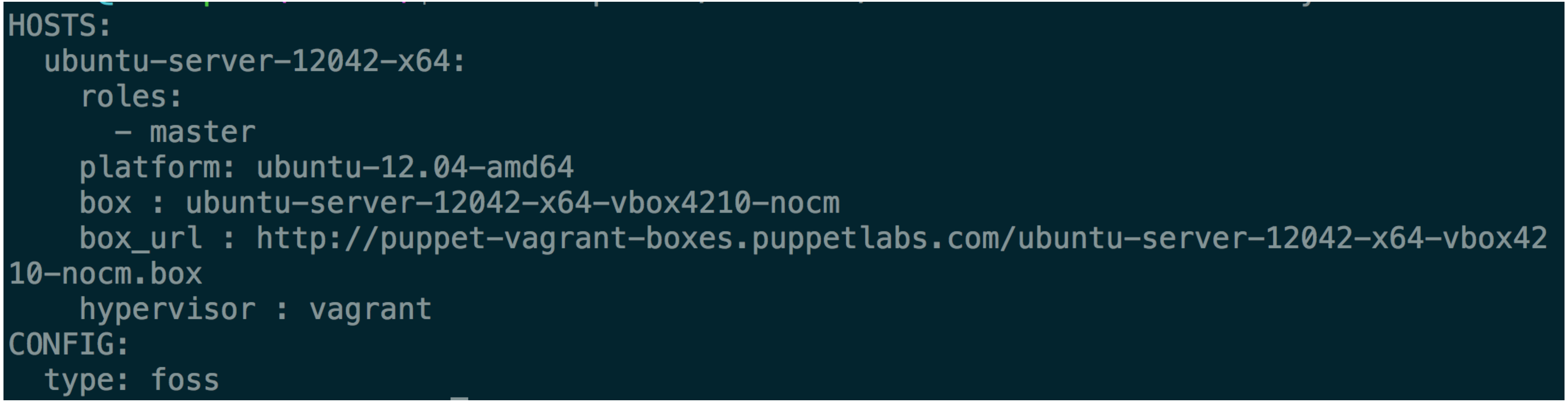
```
$ bundle exec rspec spec/acceptance
```



```
anode@eum puppetlabs-inifile (master)$ bundle exec rspec spec/acceptance
Hypervisor for ubuntu-server-12042-x64 is vagrant
Beaker::Hypervisor, found some vagrant boxes to create
created Vagrantfile for VagrantHost ubuntu-server-12042-x64
Bringing machine 'ubuntu-server-12042-x64' up with 'virtualbox' provider...
==> ubuntu-server-12042-x64: Importing base box 'ubuntu-server-12042-x64-vbox4210-nocm'
...
==> ubuntu-server-12042-x64: Matching MAC address for NAT networking...
==> ubuntu-server-12042-x64: Setting the name of the VM: ubuntu-server-12042-x64yml_ubu
ntu-server-12042-x64_1420499775611_97519
==> ubuntu-server-12042-x64: Clearing any previously set forwarded ports...
==> ubuntu-server-12042-x64: Clearing any previously set network interfaces...
==> ubuntu-server-12042-x64: Preparing network interfaces based on configuration...
    ubuntu-server-12042-x64: Adapter 1: nat
    ubuntu-server-12042-x64: Adapter 2: hostonly
==> ubuntu-server-12042-x64: Forwarding ports
```

# What’s that setfile?

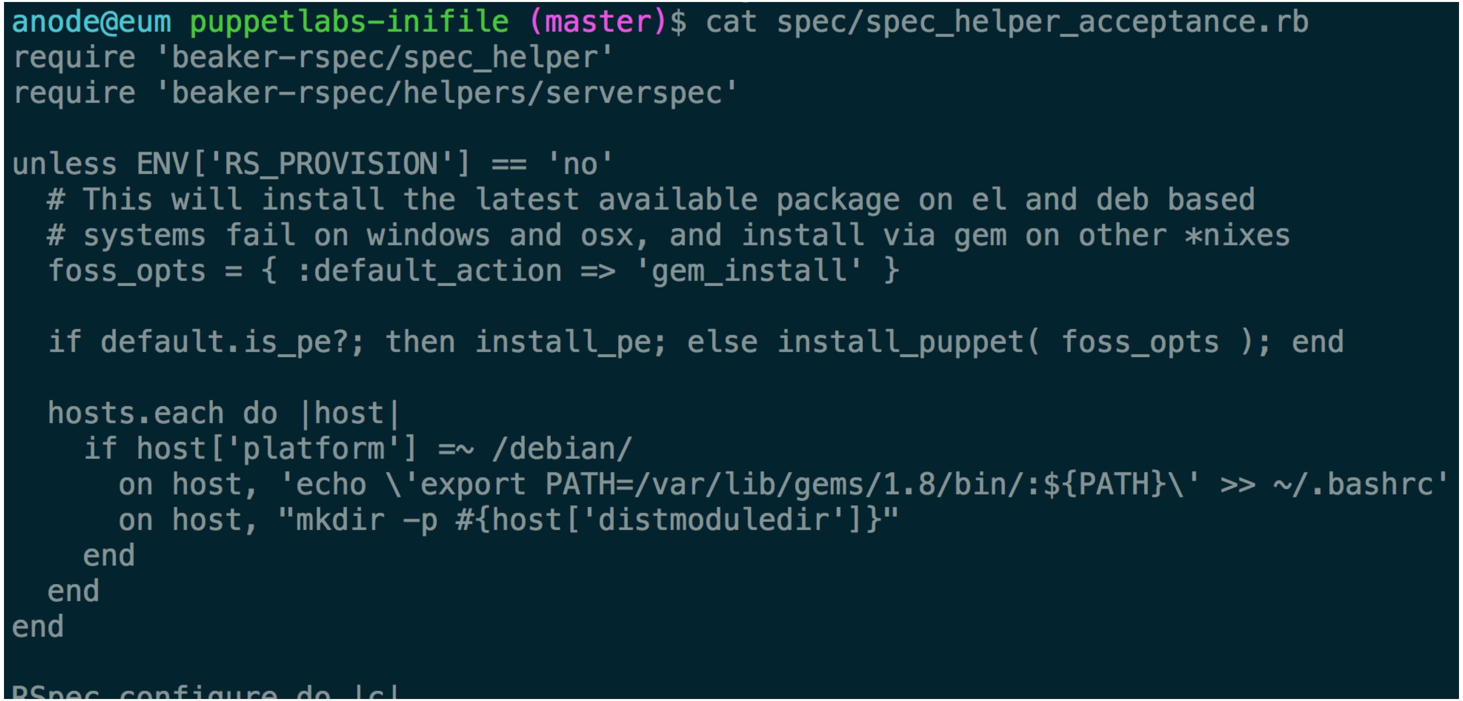
```
$ ls spec/acceptance/nodesets

$ cat spec/acceptance/nodesets/ubuntu-server-12042-x64.yml
```



```
HOSTS:
  ubuntu-server-12042-x64:
    roles:
      - master
    platform: ubuntu-12.04-amd64
    box : ubuntu-server-12042-x64-vbox4210-nocm
    box_url : http://puppet-vagrant-boxes.puppetlabs.com/ubuntu-server-12042-x64-vbox42
10-nocm.box
    hypervisor : vagrant
CONFIG:
  type: foss
```

# spec_helper_acceptance.rb

```
$ cat spec/spec_helper_acceptance.rb
```

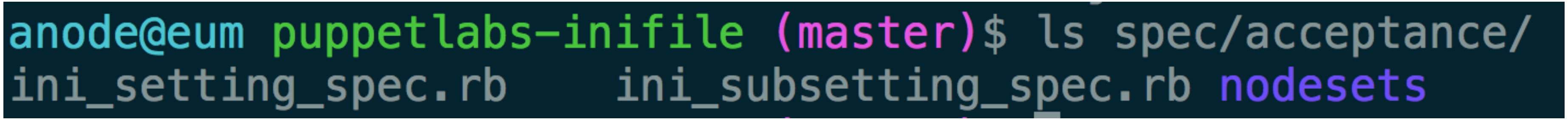


```
anode@eum puppetlabs-inifile (master)$ cat spec/spec_helper_acceptance.rb
require 'beaker-rspec/spec_helper'
require 'beaker-rspec/helpers/serverspec'

unless ENV['RS_PROVISION'] == 'no'
  # This will install the latest available package on el and deb based
  # systems fail on windows and osx, and install via gem on other *nixes
  foss_opts = { :default_action => 'gem_install' }

  if default.is_pe?; then install_pe; else install_puppet( foss_opts ); end

  hosts.each do |host|
    if host['platform'] =~ /debian/
      on host, 'echo \'export PATH=/var/lib/gems/1.8/bin/:${PATH}\' >> ~/.bashrc'
      on host, "mkdir -p #{host['distmoduledir']}"
    end
  end
end

RSpec.configure do |c|
```

KICKOFF 2015

# the tests

```
$ ls spec/acceptance
```



```
anode@eum puppetlabs-inifile (master)$ ls spec/acceptance/
ini_setting_spec.rb     ini_subsetting_spec.rb nodesets
```

KICKOFF 2015

# ENV vars for defining SUTs

```
BEAKER_set or RS_SET =>
       define hosts file as spec/acceptance/nodesets/#{BEAKER_set}

BEAKER_setfile or RS_SETFILE =>
                                          define hosts file as #{BEAKER_setfile}
```

# ENV vars for controlling SUTs

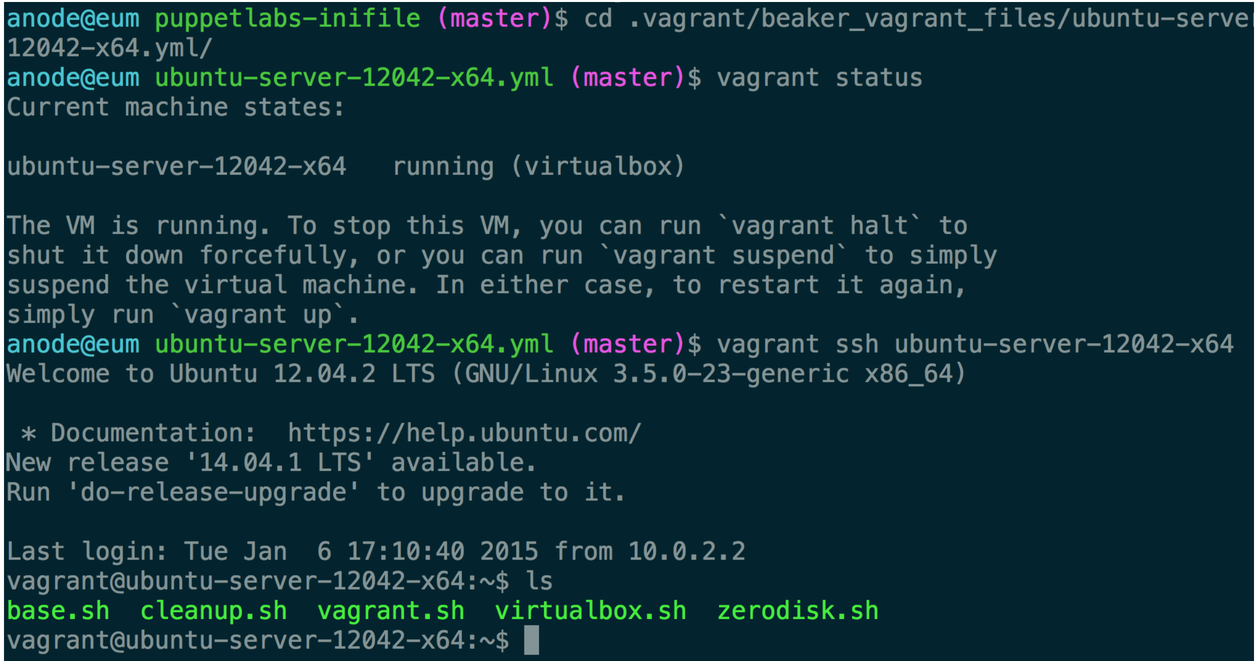
```
BEAKER_provision or RS_PROVISION =>
                        should we provision new hosts (yes/no)?

BEAKER_destroy or RS_DESTROY =>
                and should we destroy hosts after testing (yes/no)?
        should we destroy hosts only on passing tests (onpass)?
```

# additional ENV vars

```
BEAKER_keyfile or RS_KEYFILE =>
                            use this ssh key to log into test hosts

BEAKER_debug or RS_DEBUG =>
                              should we print debug during testing?
```

# SUTs for you

```
$ cd .vagrant/beaker_vagrant_files/ubuntu-server-12042-x64.yml/
$ vagrant status
$ vagrant ssh ubuntu-server-12042-x64
```



```
anode@eum puppetlabs-inifile (master)$ cd .vagrant/beaker_vagrant_files/ubuntu-serve
12042-x64.yml/
anode@eum ubuntu-server-12042-x64.yml (master)$ vagrant status
Current machine states:

ubuntu-server-12042-x64   running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
anode@eum ubuntu-server-12042-x64.yml (master)$ vagrant ssh ubuntu-server-12042-x64
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-23-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Jan  6 17:10:40 2015 from 10.0.2.2
vagrant@ubuntu-server-12042-x64:~$ ls
base.sh  cleanup.sh  vagrant.sh  virtualbox.sh  zerodisk.sh
vagrant@ubuntu-server-12042-x64:~$
```

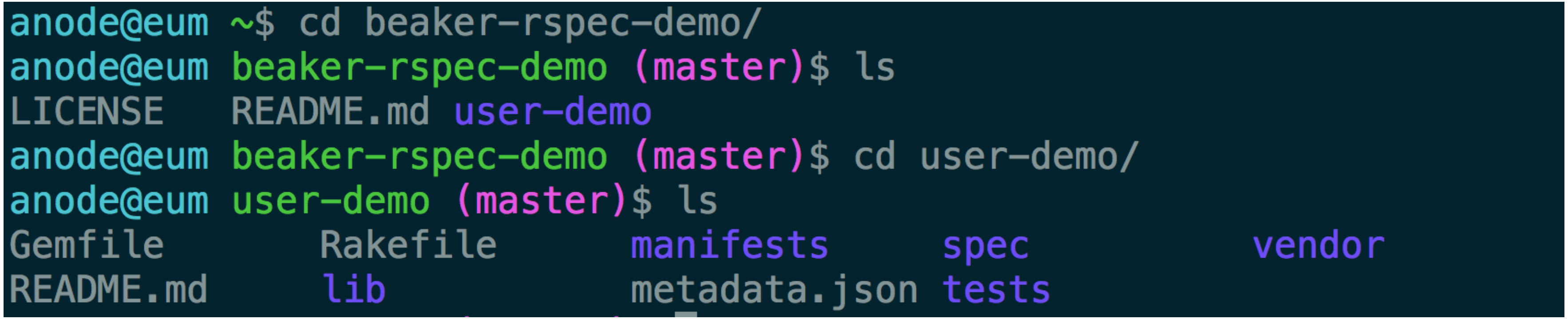
# part 2

## Steps

- clone beaker-rspec-demo
- write Beaker DSL based tests
- write Serverspec based tests
- write a combination Beaker + Serverspec test!

# clone beaker-rspec-demo

```
$ git clone https://github.com/anodelman/beaker-rspec-demo
```



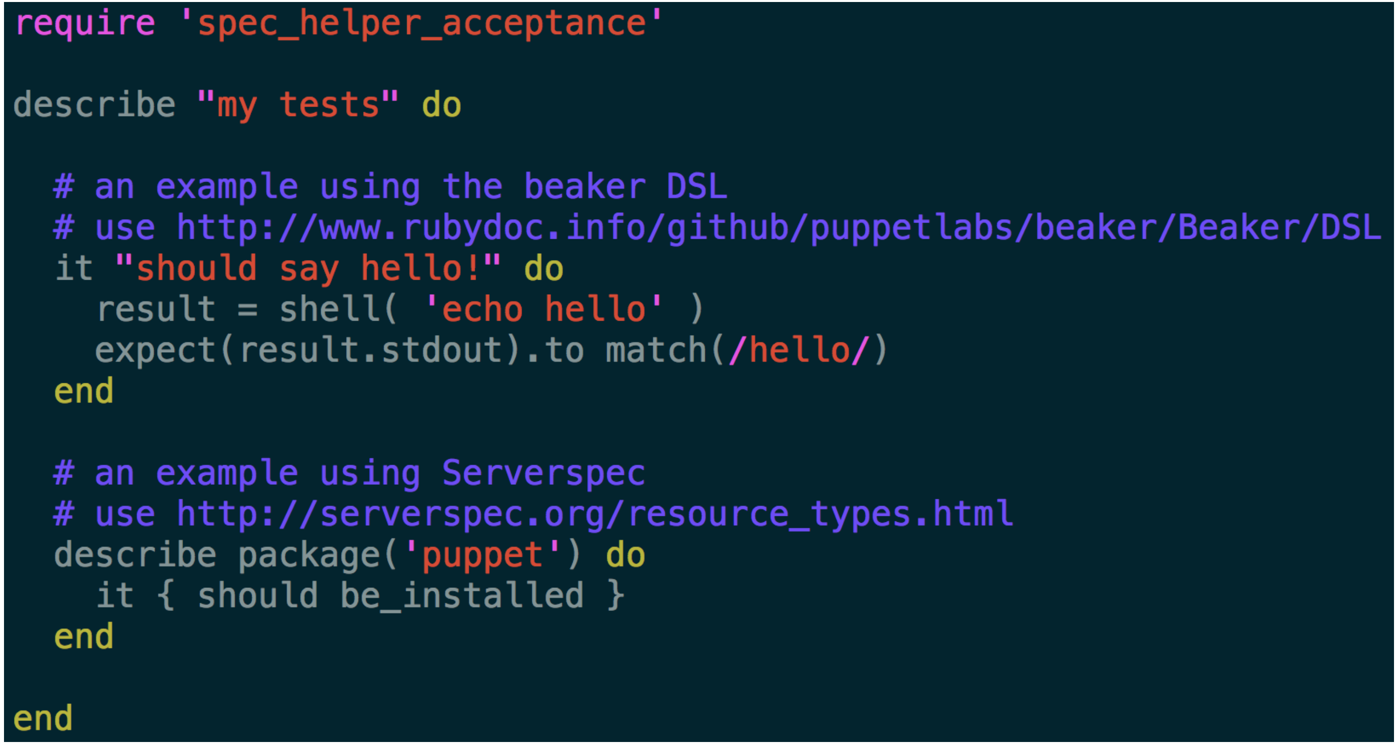
```
anode@eum ~$ cd beaker-rspec-demo/
anode@eum beaker-rspec-demo (master)$ ls
LICENSE   README.md user-demo
anode@eum beaker-rspec-demo (master)$ cd user-demo/
anode@eum user-demo (master)$ ls
Gemfile       Rakefile      manifests     spec          vendor
README.md     lib           metadata.json tests
```

# install & execute demo tests

```
$ bundle install --path vendor/bundle
$ export BEAKER_debug=true
$ export BEAKER_destroy=no
$ bundle exec rspec spec/acceptance
<--- test output here --->
```

# what’s really running here?

```
$ vi spec/acceptance/demo_spec.rb
```



```ruby
require 'spec_helper_acceptance'

describe "my tests" do

  # an example using the beaker DSL
  # use http://www.rubydoc.info/github/puppetlabs/beaker/Beaker/DSL
  it "should say hello!" do
    result = shell( 'echo hello' )
    expect(result.stdout).to match(/hello/)
  end

  # an example using Serverspec
  # use http://serverspec.org/resource_types.html
  describe package('puppet') do
    it { should be_installed }
  end

end
```

# write your own tests

# Beaker DSL Challenge (1 of 2)

1. Use *create_remote_file* to create a file called ‘demo.txt’ with contents ‘foo\nfoo\nfoo\n’
    a. Use *on* to grep the file for ‘foo’
    b. Use *on* to grep the file for ‘bar’ and catch the error

# Beaker DSL Challenge (2 of 2)

2. Use *apply_manifest* to execute these manifests, use *expect_changes* as required:
    a. user {'foo':<br>
        ensure => present,}
    b. user {'foo':<br>
        ensure => absent,}
    c. user {'root':<br>
        ensure => present,}

# Serverspec Challenge

1. Write a test that uses Serverspec to ensure that sshd is running
2. Write two tests and ensure that:
    a. ‘root’ user exists
    b. ‘foo’ user does not exist

# Together: The Final Challenge

1. Use *apply_manifest* to create a remote file (/tmp/demo.txt) with the contents ‘this is my file’
2. Use Serverspec to check that the file is present and has the correct contents.

CONGRATULATiONS
i'M PROUD OF YOU!

# resources

git

Ruby 1.9+

RubyGems

Bundler

Vagrant 1.5.2

VirtualBox

Beaker Code

Beaker-rspec Code

Beaker DSL

Pretty Beaker Junit Output

Serverspec Resource Types

Beaker-rspec Demo Code

PuppetLabs Inifile Module

Puppet Supported Modules

# contact

Alice Nodelman

Kevin Imber

Hipchat: QE-QA

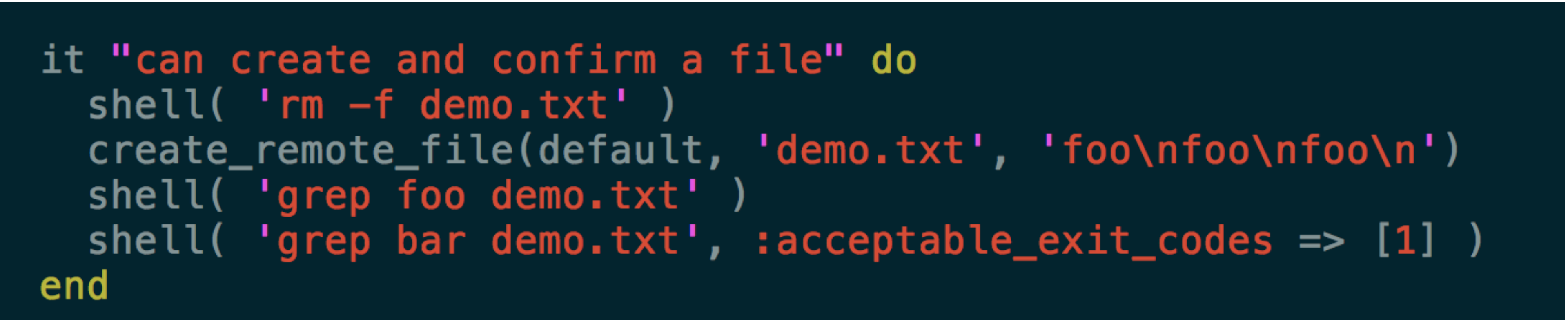
freenode irc: #puppet-dev

Jira Beaker Bug Backlog

ANSWERS

# Beaker DSL Challenge (1 of 2)

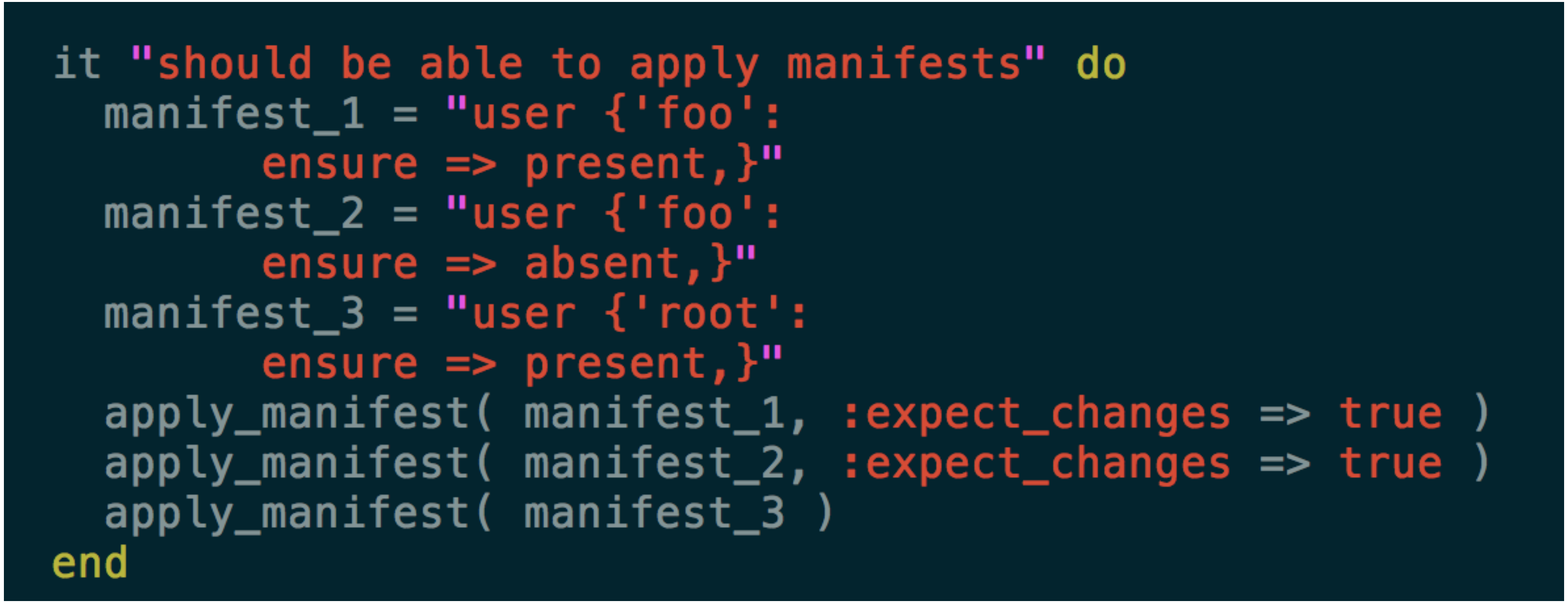
## SOLUTION



```
it "can create and confirm a file" do
  shell( 'rm -f demo.txt' )
  create_remote_file(default, 'demo.txt', 'foo\nfoo\nfoo\n')
  shell( 'grep foo demo.txt' )
  shell( 'grep bar demo.txt', :acceptable_exit_codes => [1] )
end
```

# Beaker DSL Challenge (2 of 2)

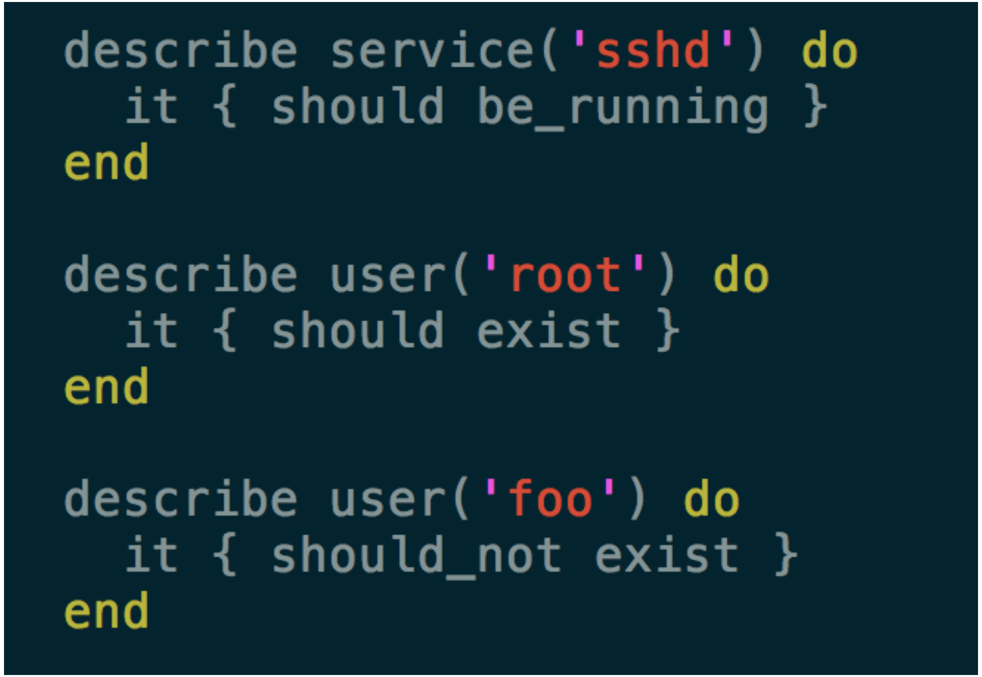
## SOLUTION



```
it "should be able to apply manifests" do
  manifest_1 = "user {'foo':
        ensure => present,}"
  manifest_2 = "user {'foo':
        ensure => absent,}"
  manifest_3 = "user {'root':
        ensure => present,}"
  apply_manifest( manifest_1, :expect_changes => true )
  apply_manifest( manifest_2, :expect_changes => true )
  apply_manifest( manifest_3 )
end
```

# Serverspec Challenge

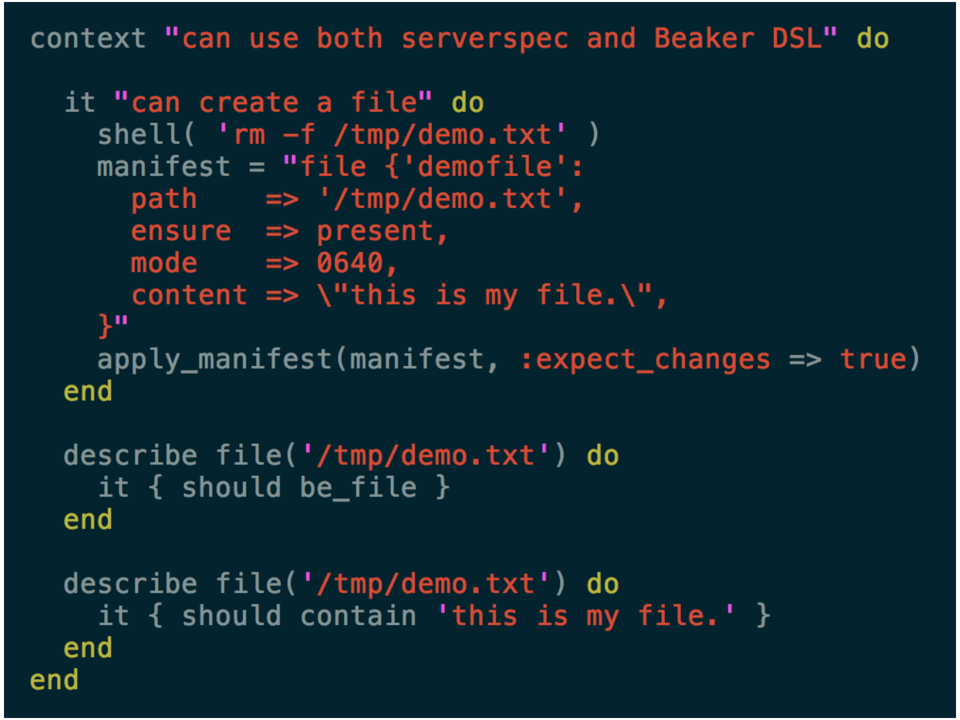
## SOLUTION



```
describe service('sshd') do
  it { should be_running }
end

describe user('root') do
  it { should exist }
end

describe user('foo') do
  it { should_not exist }
end
```

# Together: The Final Challenge

## SOLUTION



```
context "can use both serverspec and Beaker DSL" do

  it "can create a file" do
    shell( 'rm -f /tmp/demo.txt' )
    manifest = "file {'demofile':
      path    => '/tmp/demo.txt',
      ensure  => present,
      mode    => 0640,
      content => \"this is my file.\",
    }"
    apply_manifest(manifest, :expect_changes => true)
  end

  describe file('/tmp/demo.txt') do
    it { should be_file }
  end

  describe file('/tmp/demo.txt') do
    it { should contain 'this is my file.' }
  end
end
```