Puppet Configuration Management

in High Energy Physics

Outline

- Brief summary of the last HEPix Config Management WG meeting
- Converting a private module into a public module
- Unit testing, Continuous Integration (CI) and Github

Part 1

HEPIX CONFIG MEETING

Last Configuration WG meeting

- Roundtable of many institutes: https://twiki.cern.ch/twiki/bin/view/HEPIX/VideoMeeting14112013
- Some sites have their complete configuration in Puppet
- Many private modules
- We need documentation and a list of useful Puppet modules

Used puppet modules

- We exchanged used modules on hepix-config-wg@desy.de
- A long list is available on http://kreczko.web.cern.ch/kreczko/Config_Management/Puppet_modules_in_use.txt
- GitHub groups: cernops, cvmfs, HEP-Puppet, oxford-physics (and many more outside HEP)

Private modules

- Private modules are used at the sites
- No manpower to convert them into public modules
- Some modules are not “polished” enough to be made public
- Could fill in the ‘gaps’

Private modules

- Private modules are used at the sites
- No manpower to convert them into public modules
- Some modules are not “polished” enough to be made public
- Could fill in the ‘gaps’

There is no such thing!

You ~~can~~ should make a module public from day 1!

Documentation

- Ideally the code is the documentation
  - This is very difficult to do and therefore not the norm
  - Most importantly: Usage must be clear
  - But even then it is only light documentation
- Generally documentation is written on-demand (create demand!)

Part 2

CONVERTING A PRIVATE MODULE

Converting a private module

- Big current problem in the HEP Puppet community are private modules
  - Undermine the effort sharing knowledge
  - Lead to duplication of effort (wasted wo-/manpower)
  - Are more prone to bugs (less people use/see)
  - Converting them usually takes a lot of time
- The following is an example of what it takes to convert such a module and what to avoid in a new module

Nagios module

- The Nagios module was donated by Bristol's IT services
- The following is a summary of what private details I found **plus** what I've seen before
- Generally private information is included in the modules out of convenience

Private data

IP addresses/FQDN

Custom paths

certificates

Private module

hostgroups

passwords

firewall rules

Munge keys!

Removing private data

- First, you need to find it
  - There is no ultimate regex expression
  - Worst case: read ALL the code
- The most complete way to remove private data is to start the module from scratch!
  - Of course it is **very** useful to reuse code
  - But beware of copy & paste!

Part 3

TESTING THE CODE

Why test your code?

Missing files

typos

Syntax errors

Mistakes happen

Missing dependencies

Wrong implementation

Why test your code?

New parameter has bad default

Do not want to trash a production system

Silent errors

Template incorrect

Why test your code?

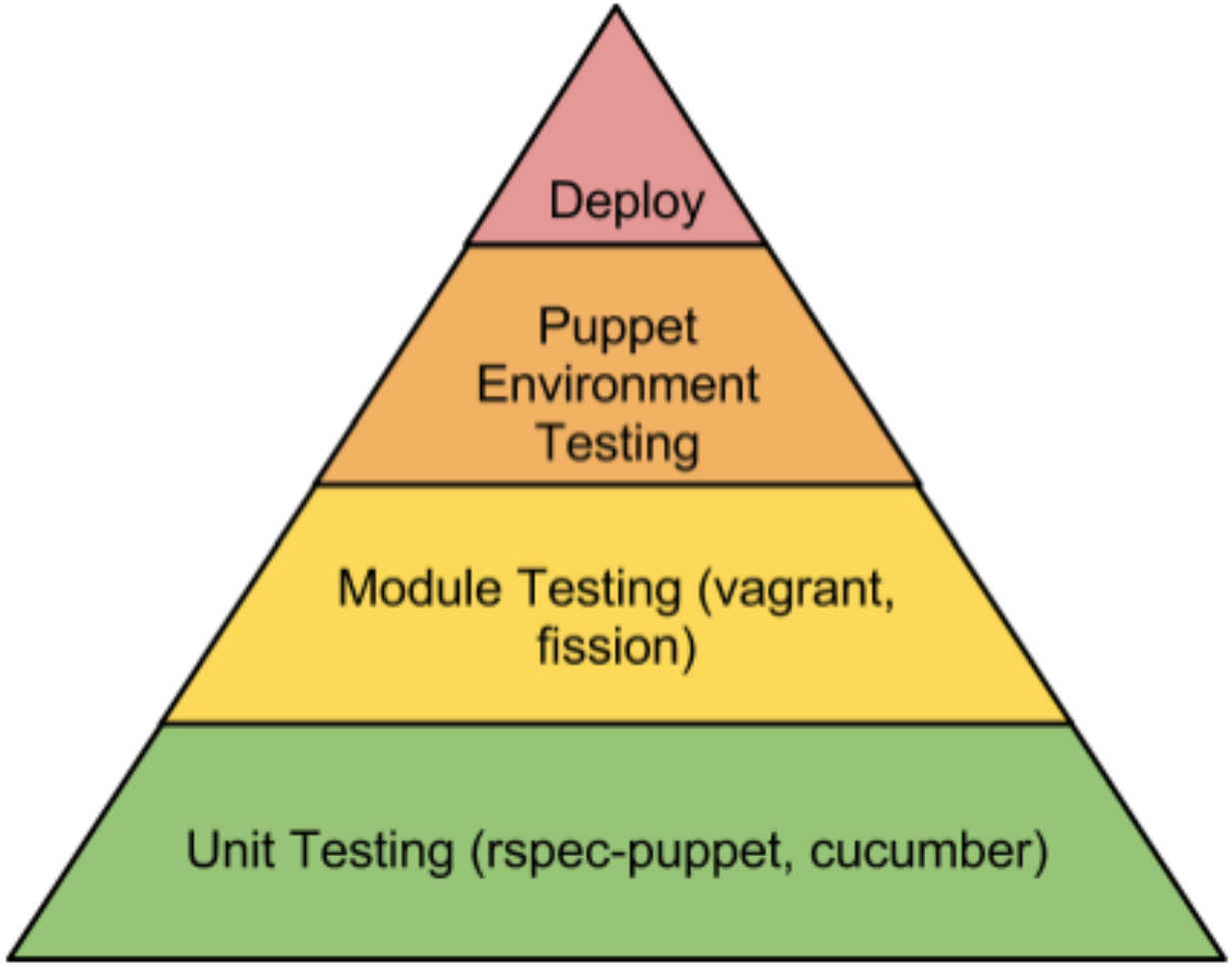
```
root@testnode >> puppet agent -t –noop  #is not sufficient
```

Only useful if testnode is cleaned for each test

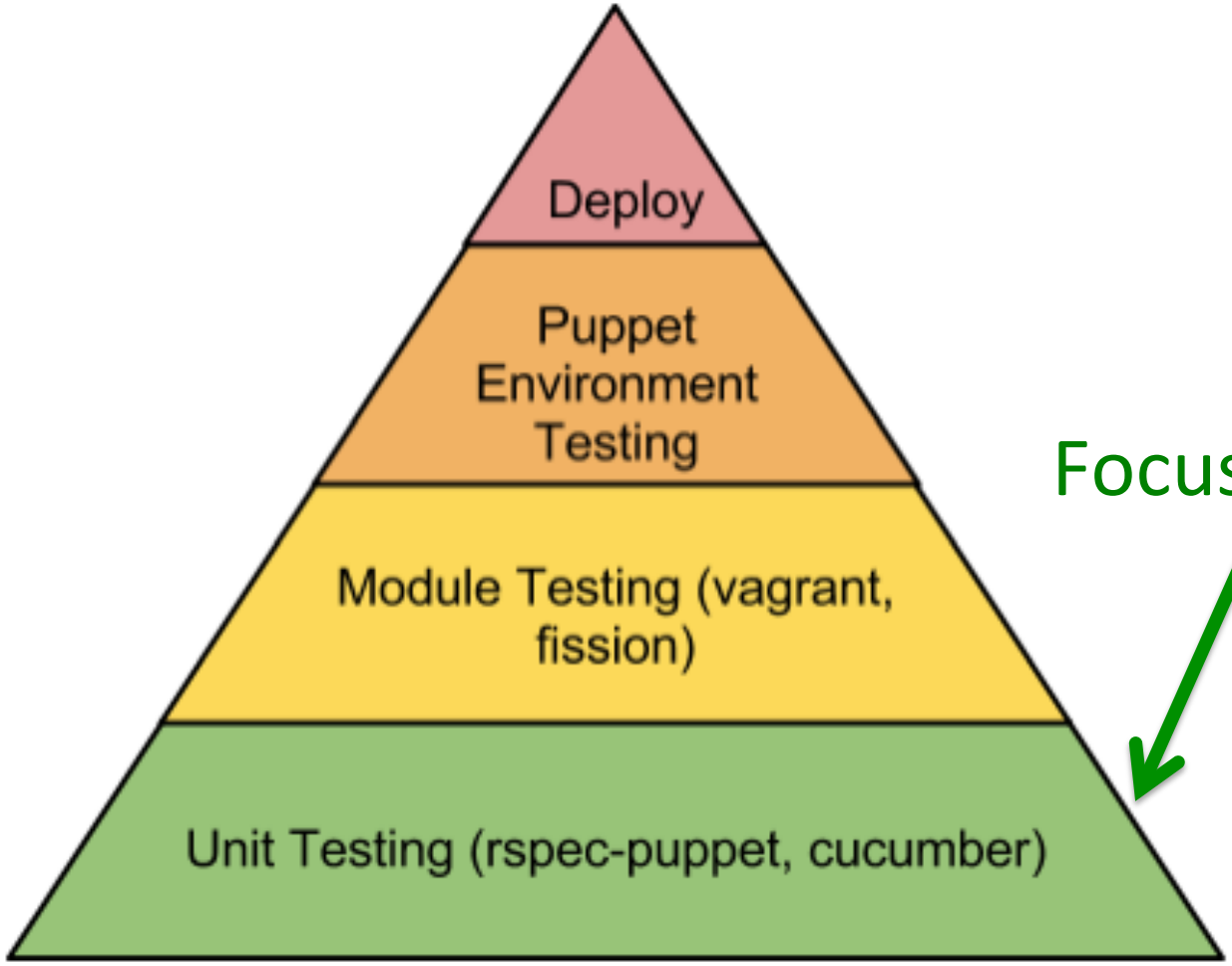
Testing your Puppet code

- Lots of information available:
  - http://puppetlabs.com/blog/test-driven-development-with-puppet/
  - http://puppetlabs.com/blog/the-next-generation-of-puppet-module-testing
  - http://bombasticmonkey.com/2012/03/02/automatically-test-your-puppet-modules-with-travis-ci/
  - https://github.com/puppetlabs/puppetlabs-apache/blob/master/spec/classes/apache_spec.rb

The testing pyramid



The testing pyramid



Focus today

How to test?

- Locally: with rake tasks (lint, unit tests)
- Centrally (remotely): with rake tasks

How to test?

- Locally: with rake tasks (lint, unit tests)
- Centrally (remotely): with rake tasks

But

What if you forget to run them?

Nobody else sees the results

How to test?

- Locally: with rake tasks (lint, unit tests)
- Centrally (remotely): with rake tasks

Better

Automatic and (can be) visible for all

But

Needs more setup and a test server

What is needed?

- Server: travis-ci.org (free and integrated with github)
- Configuration:
  - .travis.yml
  - Gemfile
  - .fixtures.yml
  - .nodeset.yml
- Tests:
  - Spec/<classes/functions/etc>/*.rb

Travis-ci.org

HEP-Puppet/nagios

build passing

Puppet module to set up Nagios monitoring for your infrastructure

Current | Build History | Pull Requests | Branch Summary

Build	28	Commit	d0b417b (development)
State	Passed	Pull Request	#8 Improvements and fixes
Finished	less than a minute ago	Author	Luke
Duration	5 min 38 sec	Committer	Luke
Message	Improvements and fixes: - check kernel plugin: correcting capitalisation and RedHat <-> Debian for one case - puppet plugin: removing checking db test - adding vhosts to server config and adding parameter passthrough		

Links to console output

Build Matrix

Job	Duration	Finished	Ruby	Gemfile	ENV
28.1	3 min 2 sec	less than a minute ago	1.8.7	Gemfile	PUPPET_VERSION=2.7.23
28.2	2 min 36 sec	less than a minute ago	1.8.7	Gemfile	PUPPET_VERSION=3.3.2

Travis-ci.org

- On the github side (pull request) you see first

- And then either

- Or

All examples are taken from HEP-Puppet nagios module

.travis.yml

- Instructions for travis-ci.org

```
language: ruby

rvm:
  - 1.8.7

script: "rake spec SPEC_OPTS='--color --format documentation'"

branches:
  only:
    - master
    - development

notifications:
  email: false

gemfile: Gemfile
```

```
env:
  - PUPPET_VERSION=2.7.23 # latest 2.7; PE 2.8.0+
  - PUPPET_VERSION=3.3.2 # latest 3.3;

before_install:
  - travis_retry gem update --system 2.1.11
  - travis_retry gem install bundler --pre
  - gem --version
  - bundle –version
```

.nodeset.yml

- OS definitions for the test

```
---
default_set: 'centos-64-x64'
sets:
  'centos-59-x64':
    nodes:
      "main.foo.vm":
        prefab: 'centos-59-x64'
  'centos-64-x64':
    nodes:
      "main.foo.vm":
        prefab: 'centos-64-x64'
  'ubuntu-server-12042-x64':
    nodes:
      "main.foo.vm":
        prefab: 'ubuntu-server-12042-x64'
```

.fixtures.yml

- Module dependencies

```
fixtures:
  repositories:
    "apache": "https://github.com/puppetlabs/puppetlabs-apache.git"
    "concat": "https://github.com/puppetlabs/puppetlabs-concat.git"
    "stdlib": "https://github.com/puppetlabs/puppetlabs-stdlib"
    "firewall": "https://github.com/puppetlabs/puppetlabs-firewall"
    "grid_repos": "https://github.com/HEP-Puppet/grid_repos"
  symlinks:
    "nagios": "#{source_dir}"
```

Gemfile

- Gem dependencies

```
source 'https://rubygems.org'

puppetversion = ENV.key?('PUPPET_VERSION') ? "= #{ENV['PUPPET_VERSION']}" : ['>= 2.7']
gem 'puppet', puppetversion
gem 'puppet-lint', '>= 0.3.2'
gem 'puppetlabs_spec_helper', '>= 0.1.0'
```

- Files that are not mentioned:
  - spec/spec_helper.rb
  - spec/spec.opts

Nagios_spec.rb

```
1  require "#{File.join(File.dirname(__FILE__),'..','spec_helper.rb')}"
2
3  describe 'nagios' do
4
5    let(:title) { 'nagios' }
6    let(:node) { 'testing.phy.bris.ac.uk' }
7    let(:facts) { {
8        :ipaddress => '10.13.37.100',
9        :processorcount => 1,
10       :osfamily => 'RedHat',
11       :operatingsystem => 'Redhat',
12       :operatingsystemrelease => '6.4',
13       :concat_basedir         => '/dne',
14     } }
15
```

Most facter variables are empty!
Need so set them!

Nagios_spec.rb

```
describe 'Test standard installation on RedHat (server)' do
  let(:params) { {:is_server => true } }
  # packages
  it { should contain_package('nagios').with_ensure('installed') }
  it { should contain_package('apache').with_ensure('installed') }
  it { should contain_package('pnp4nagios').with_ensure('installed') }
  it { should contain_package('nagios-plugins-nrpe').with_ensure('installed') }
  it { should contain_package('nsca').with_ensure('installed') }
  # services
  it { should contain_service('nagios').with_ensure('running') }
  it { should contain_service('nagios').with_enable('true') }
  it { should contain_service('httpd').with_ensure('running') }
  it { should contain_service('httpd').with_enable('true') }
  # files
  it {should contain_file('resource.cfg').with({
      'owner' => 'root',
      'group' => 'nagios',
      'mode' => '0640',
    })}
  it {should contain_file('nagios.cfg').with({
      'owner' => 'root',
      'group' => 'nagios',
      'mode' => '0640',
    })}
  it {should contain_file('nsca.cfg').with({
      'owner' => 'root',
      'group' => 'root',
```

SUMMARY

Summary

- We are slowly reaching critical mass
  - Each meeting more contributors! That's fantastic!
- Private modules don't safe time – it is the opposite
- Unit testing is very important if we want to be serious about production quality modules
- Further down the line we should think about test-site deployment

The end

QUESTIONS?

References

- Nagios module:

  https://github.com/HEP-Puppet/nagios

- Grid_repos module:

  https://github.com/HEP-Puppet/grid_repos

- HEP-Puppet/nagios on travis-ci.org:

  https://travis-ci.org/HEP-Puppet/nagios

References

- Config Management meeting 14.11.2013: https://indico.cern.ch/conferenceDisplay.py?confId=283028
- Hep-Puppet: https://github.com/HEP-Puppet
- Twiki: https://twiki.cern.ch/twiki/bin/view/HEPIX/ConfigManagement