Testing your puppet code

Julien Pivotto

Libre Software Meeting 2013

July 15, 2013

Julien Pivotto

- sysadmin @ inuits
- open-source defender for 7+ years
- devops believer
- @roidelapluie on twitter/github

Infrastructure as Code

- Keep your environments under SCM
- Overview of complete environments
- Reduce the deployment time

Keep all environments the same

http://www.flickr.com/photos/bobvietnam/4828291896/

Packaging with FPM

- Ruby gem
- package a directory (and much more)
- Support .deb, .rpm
- Package the code with several prefixes
- /etc/puppet/environments/infradev
- /etc/puppet/environments/uat

Vagrant

- Create virtual machines
- Provision them
- Destroy & recreate

Vagrant

- Chef, scripts, puppet, ...
- Backend: Virtualbox, KVM, ...
- A lot of baseboxes available
- http://vagrantup.com

Vagrant

- Local testing
- The same environment as the target

Puppet environments

- Multiple environments
- The same tree for all the environments
- Pushing changes to UAT/prod on-demand
- Small changes vs big releases

Hiera

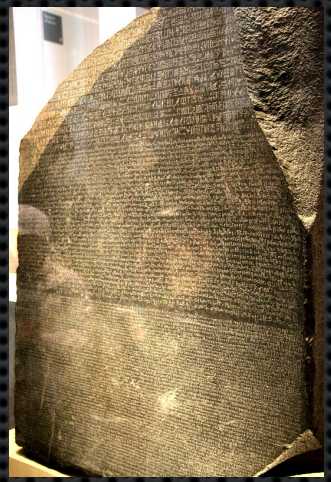
- Storing the data in Hiera(-gpg)
- Usernames, password, IP addresses
- Hiera is made to be structured
- Using one hiera repo for all the environments
- Using Hiera in your manifests, not in your modules

Hiera tree

- %{environment}/%{hostname}
- %{environment}/common
- infradev/www45.yaml
- infradev/common.yaml

Keeping clean puppet modules

http://www.flickr.com/photos/aurelie_solenne/8340968061/



- Make them readable
- Make them reusable and sharable
- Don't puppetize everything
- User generated content is not puppetized

Use the right structure for your modules

- Package, config, service
- module::package, module::config, module::service
- Parameterized classes

http://www.slideshare.net/PuppetLabs/modern-module-development-ken-barber-2012-edinburgh-puppet-camp

Distribution-agnostic puppet modules

- You don't have to support all the distros
- Adding support for another distro should be easy

```
$config_dir = $configroot ? {
  undef   => $::operatingsystem ? {
    /Debian|Ubuntu/ => '/etc/apache2',
    /CentOS|RedHat/ => '/etc/httpd',
    default         => '/etc/httpd',
  },
  default => $configroot,
}
```

Puppet function

- The fail function prevents catalog to be applied
- The notify function prints a warning

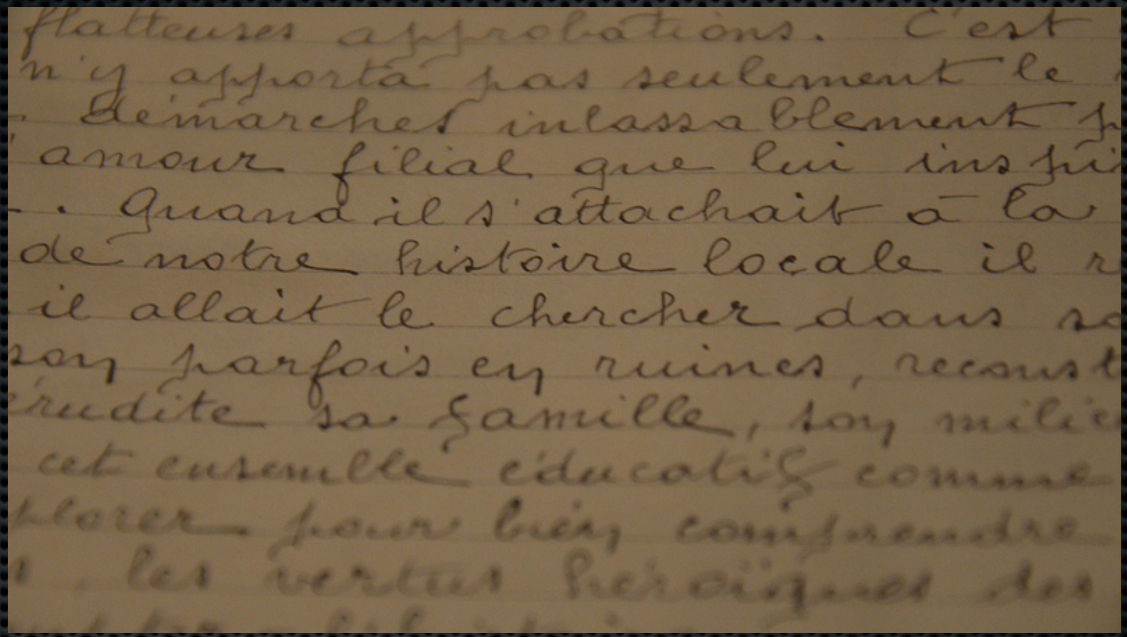
```
if (!$leftsubnet) and (!$leftsubnets) {
 fail('$leftsubnets and $leftsubnet both empty')
}
```

Puppet parser

- Included in puppet
- Validating the syntax
- `puppet parser validate init.pp`
- `find . -name '*.pp' -exec puppet parser validate + ;`

Puppet lint

http://www.flickr.com/photos/voyages-provence/8127668094/



- Follow the puppet style guide
- Two-space soft tab
- align fat comma arrows (=>) within blocks of attributes
- http://docs.puppetlabs.com/guides/style_guide.html

Cucumber puppet

- Write scenarios
- Easy to read (full sentences)
- Use your manifests
- Need some tricks to work with Puppet 3
- Discontinued

Cucumber example

Cucumber

```
Feature: General catalog policy
  In order to ensure applicability of a host's catalog
  As a manifest developer
  I want all catalogs to obey some general rules

  Scenario Outline: Compile and verify catalog
    Given a node specified by "features/yaml/<hostname>."
    When I compile its catalog
    Then compilation should succeed
    And all resource dependencies should resolve

    Examples:
      | hostname  |
      | localhost |
```

rspec-puppet

- Check what is the behaviour of puppet
- Separate tests per modules
- Add context, facts, ...
- Test custom functions, hosts, manifests, ...

rspec-puppet

Start with rspec puppet

```
gem install rspec-puppet
gem install puppet
cd my-module
rspec-puppet-init
```

rspec-puppet

spec/defines/connection_spec.rb

```
require 'spec_helper'
describe 'openswan::connection' do
  describe 'should require rightsubnet or rightsubnets' do
    let(:title) { 'foobar' }
    let (:params) { {
      :ike         => 'aes256-sha1;modp1024',
      :esp         => 'aes256-sha1;modp1024',
      :leftsubnet => '8.8.5.5',
      :right       => '84.54.105.5',
      :left        => '68.65.98.6',
      :foreignip   => '45.25.5.5',
      :localtestip => '82.8.8.8', } }
      it do
        expect {
          should contain_file("/etc/ipsec.d/foobar.conf")
        }.to raise_error(Puppet::Error, /$rightsubnets and $rightsubnet cannot be both empty/)
      end
  end
end
```

rspec-puppet

Second example

```
require 'spec_helper'
describe 'apache', :type => :class do
  let (:facts) { {
    :operatingsystem => 'CentOS',
    :osfamily => 'RedHat',
  } }
  describe 'without parameters' do
    it { should create_class('apache') }
    it { should include_class('apache::service') }
    it { should contain_apache__listen('80') }
    it { should contain_apache__namevhost('80') }
  end
end
```

rspec-puppet

- should, should_not
- should contain_package
- contain_foo__bar('baz') (for foo::bar)

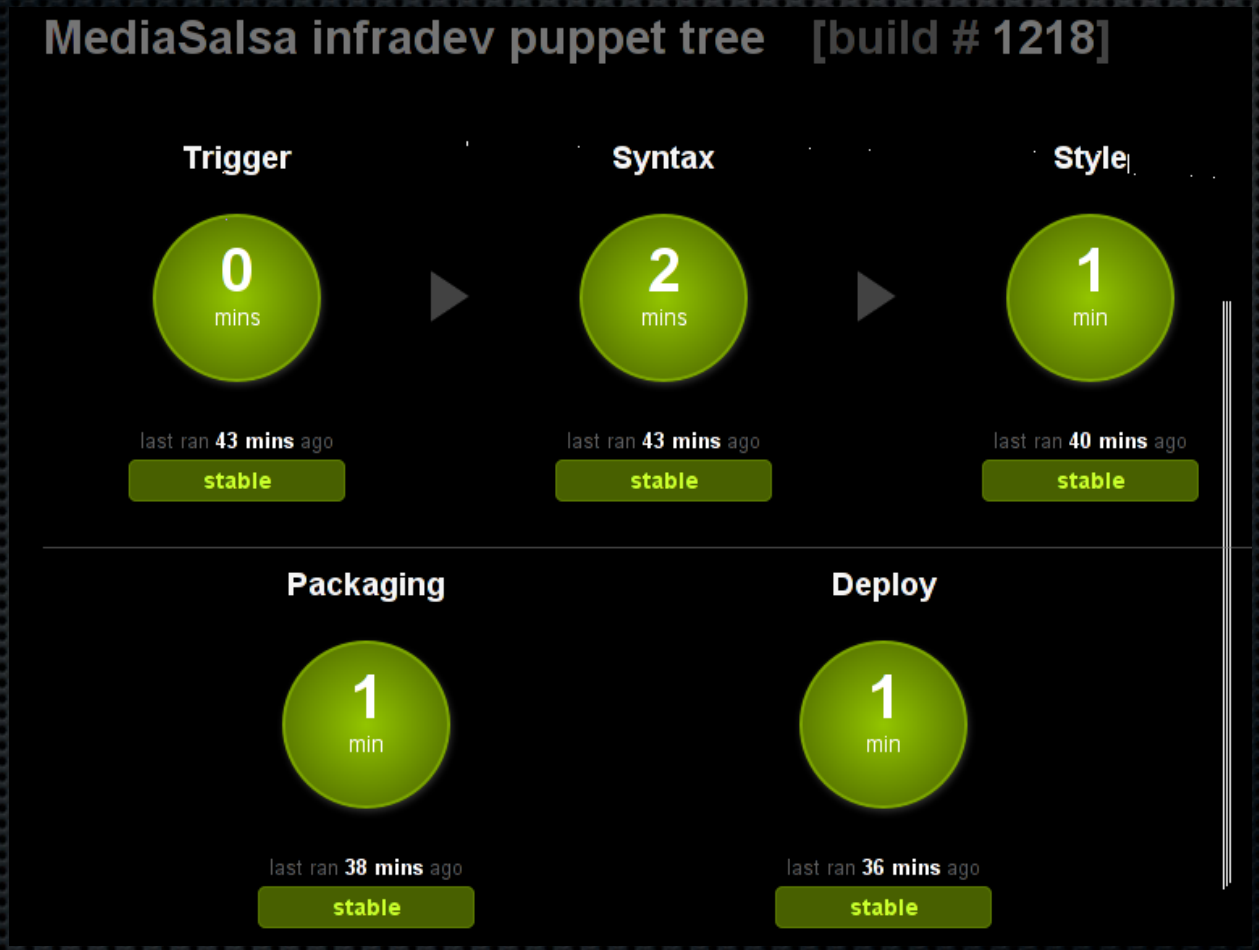
Integration with jenkins

- Pulling, testing and deployments
- Push-Test-Package-Deploy
- Continuous integration
- Continuous delivery

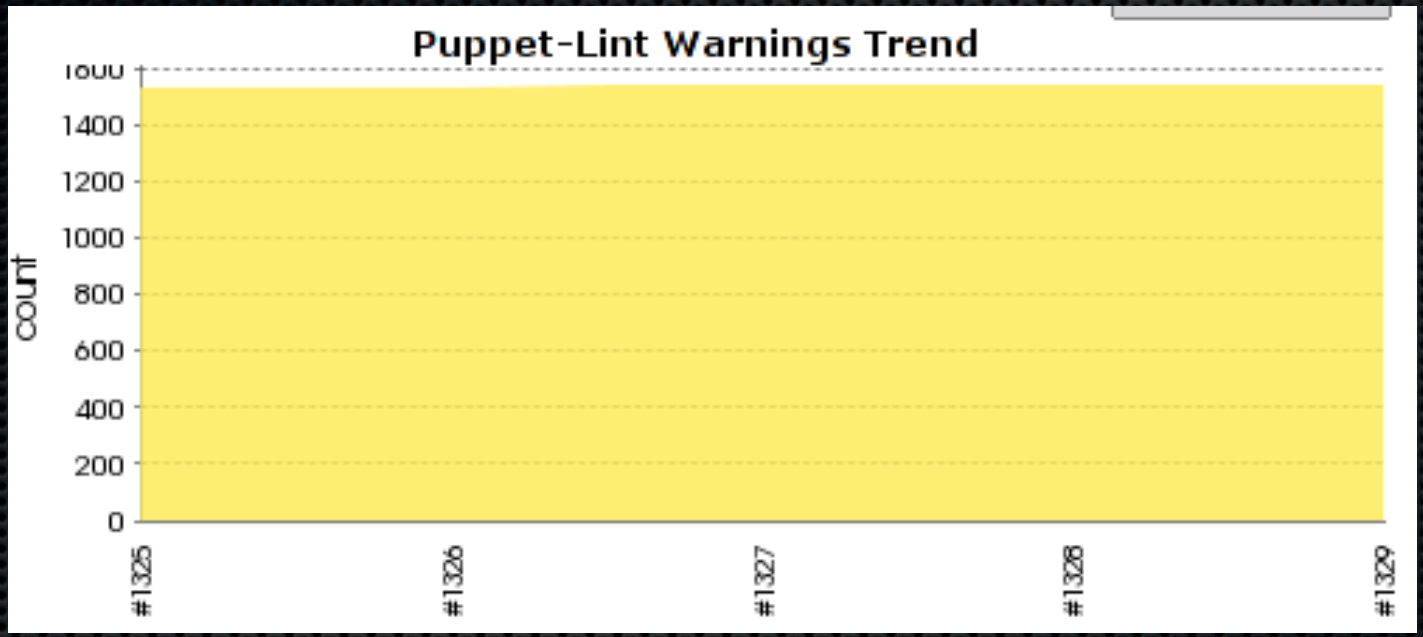
Jenkins pipelines

- Build pipelines
- Overview of what happens
- Getting notified about what failed
- Promoted build plugin

Jenkins pipelines



Advantages of CI

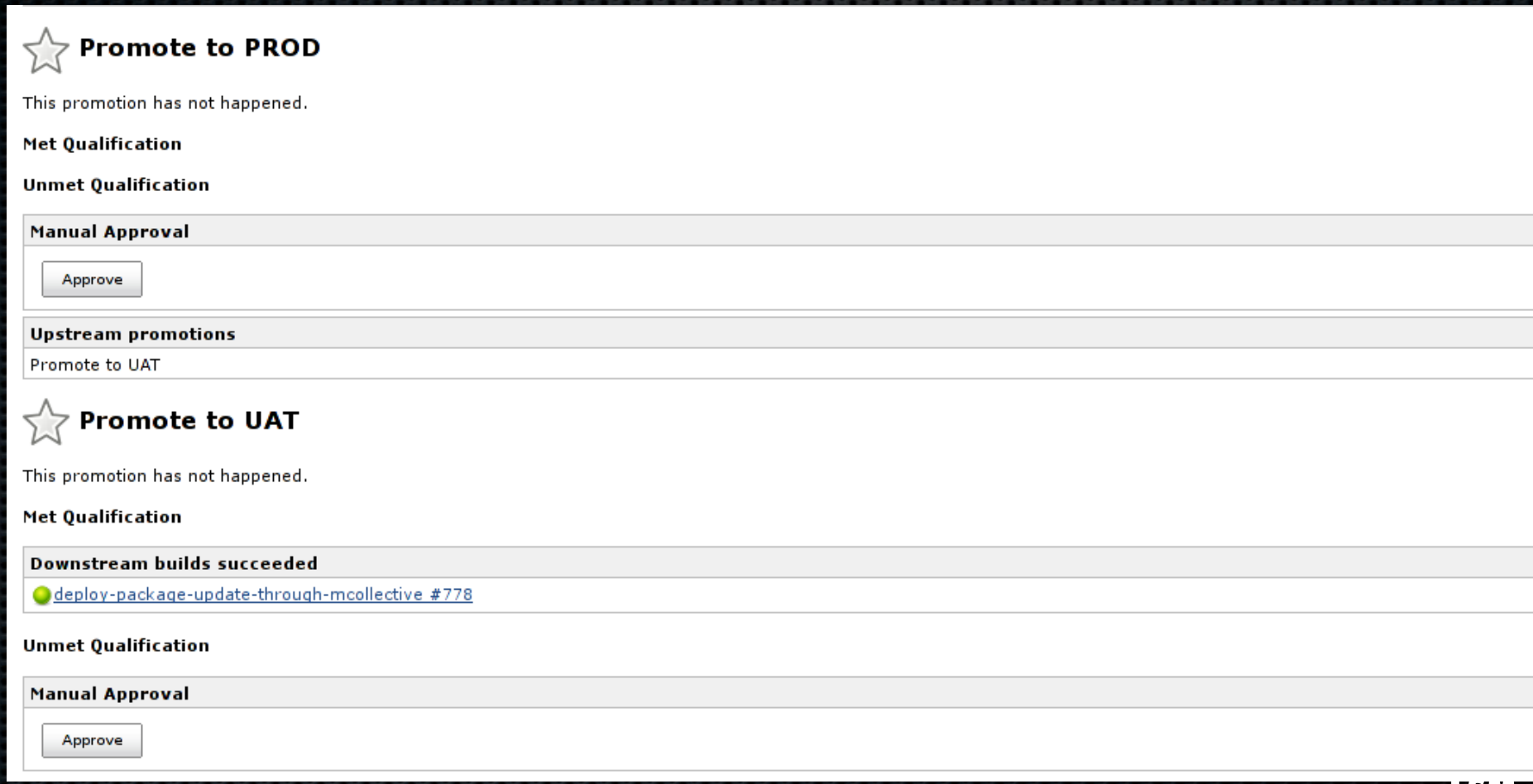


- You trust your code
- Reproducability
- You get metrics: number of warning, . . .
- You have a backlog
- It is easy!

Promotions

- Provides buttons you can click
- Trigger actions
- deploy to other environments
- Get a mail with the changes
- Have a log of who deployed

Promotions


Promote to PROD
This promotion has not happened.
Met Qualification
Unmet Qualification
Manual Approval
Approve
Upstream promotions
Promote to UAT
Promote to UAT
This promotion has not happened.
Met Qualification
Downstream builds succeeded
deploy-package-update-through-mcollective #778
Unmet Qualification
Manual Approval
Approve

Homework

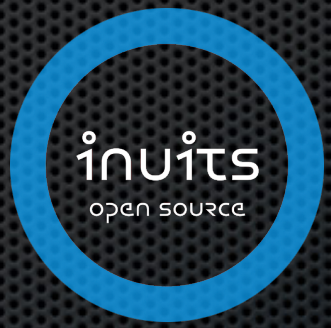
- Integrating tests with git hooks
- Integrating tests with VI
- github.com/philandstuff/fizzgig

Conclusion

- Use nice & simple Puppet modules
- Continuous integration
- Multiple environments
- Readability & reusability
- Tools exist and work together

Contact

Julien Pivotto
julien@inuits.eu
@roidelapluie



INUITS bvba
Duboisstraat 50
2060 Antwerp
Belgium
+32 473 441 636
https://inuits.eu