

Testing The Foreman

“Has Jenkins melted again...?”



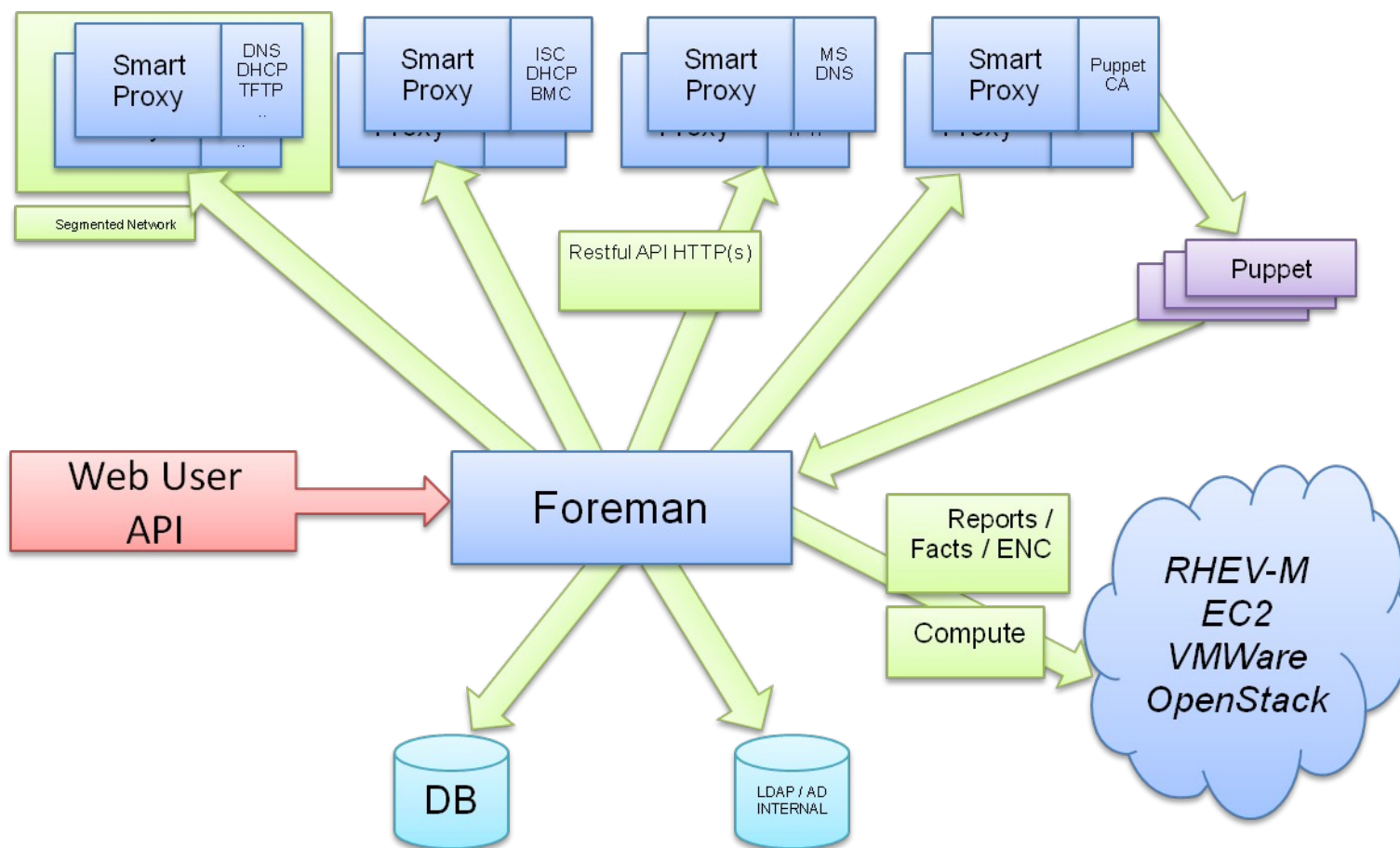
Greg Sutcliffe

Red Hat

FOSDEM 2014



Why is Foreman hard to test?



Why is Foreman hard to test?

- Complex project involving multiple subsystems
 - DNS, DHCP, TFTP, Compute Resources, ConfigMgmt
- Standard webapp axes:
 - Ruby version, database choice
- Unit tests can only get you so far
- Integration tests will only exercise your UI
- Package tests
 - Different OSs handle things in different ways

Talk Agenda

- Life cycle of a PR
 - Jenkins/GitHub integration
 - Test jobs
 - Plugin tests
 - Package build
 - Package tests
- Out of band testing
 - ~40 executors, ~50 jobs
 - ~20 jobs / hour

| | | |
|--|--|---|
| Building test_katello_core #922 | | ✖ |
| slave01.rackspace.theforeman.org | | |
| 1 | Building test_develop_pull_request = sqlite3,3.0,1.8.7 #5956 | ✖ |
| 2 | Building test_develop_pull_request = mysql,3.0,2.0.0 #5958 | ✖ |
| 3 | Building test_develop_pull_request = sqlite3,3.0,1.8.7 #5957 | ✖ |
| 4 | Building test_develop_pull_request = sqlite3,3.0,1.8.7 #5954 | ✖ |
| Building test_katello_pull_request #898 | | ✖ |
| Building test_katello_pull_request #899 | | ✖ |
| slave02.rackspace.theforeman.org | | |
| 1 | Building test_develop_pull_request = sqlite3,3.0,1.8.7 #5963 | ✖ |
| 2 | Building test_develop_pull_request = sqlite3,3.0,1.8.7 #5961 | ✖ |
| 3 | Building test_develop_pull_request = sqlite3,3.0,1.9.3 #5964 | ✖ |
| 4 | Building test_develop_pull_request = mysql,3.0,2.0.0 #5963 | ✖ |
| Building test_develop_pull_request #5954 | | ✖ |
| Building test_develop_pull_request #5955 | | ✖ |
| Building test_develop_pull_request #5956 | | ✖ |
| Building test_develop_pull_request #5957 | | ✖ |
| Building test_develop_pull_request #5958 | | ✖ |

Jenkins & GitHub

- Team members get automatic PR tests
 - They can also start tests on other PRs by commenting “[test]”
- Computationally expensive
 - Ruby/DB matrix job
 - Unit/functional tests
 - Integration tests on Psql **only**
 - Lots of PRs
- No automated merges

● Waiting to hear about 38f5f3a — Running build ([Details](#))

● 38f5f3a

✓ All is well — Job result: SUCCESS ([Details](#))

✓ 4179e3c

✗ Failed — Job result: FAILURE ([Details](#))

✗ fad6501

| Configuration Matrix | 1.8.7 | 1.9.3 | 2.0.0 |
|----------------------|-------|-------|-------|
| postgresql | ● | ● | ● |
| sqlite3 | ● | ● | ● |
| mysql | ● | ● | ● |

Merged to develop

- Full test suite is run after every commit
 - Exactly the same test setup as for PRs
 - Should never fail due to PR tests
 - Updated gems often break this theory
- IRC notification of success/failure
- Success triggers packaging & plugin tests

<jenkins-foreman Project test_develop build #520: SUCCESS
in 29 min: http://ci.theforeman.org/job/test_develop/520/

Plugin Tests

- Rails Engines
 - Separate gem loaded by Rails on startup
 - Usually tested via “dummy_app” in the engine
 - Foreman engines extend “rake test” in core

```
$ cat lib/templates.rake
Rake::Task[:test].enhance do
  Rake::Task['test:templates'].invoke
end
```

```
$ cat test/test_plugin_helper.rb
require 'test_helper'

FactoryGirl.definition_file_paths <<
File.join(File.dirname(__FILE__),
'factories')

FactoryGirl.reload
```

Package Building in Jenkins

- RPMs
 - Built using Mock, Tito, and Koji to publish
 - Repoclosure is used to verify all the dependencies
- DEBs
 - Built using PBuilder, and published with Freight
- Both are built using Jenkins script executors
- RPMs are tested **before** being published
- DEBs are tested **after** (we'd like to fix this)
- Success triggers downstream package testing

Package Testing - Bats

- Bash test framework
- Exit status
- Syntactic sugar
- `set -e`

```
@test "the truth" {  
    exit 0  
}  
  
@test "check web app is up" {  
    curl -sk $URL | grep -q  
    login  
}
```

```
tPackageInstall() {  
    yum -y install "$1"  
}  
  
@test "install foreman" {  
    tPackageInstall foreman  
}  
  
@test "stop puppet agent (if  
installed)" {  
    tPackageExists "puppet" ||  
    skip "Puppet not installed"  
    service puppet stop  
}
```

Package Testing - Results

1 files 17 tests, 13 ok, 0 not ok, 4 skipped, 0 Bail Out!

File: [fb-install-foreman.bats.out](#)

```
ok 1 # skip (Puppet package not installed)
ok 2 # skip (Puppet not installed)
ok 3 # skip (Puppet not installed)
ok 4 # skip (EPEL not required)
ok 5 configure repository
ok 6 install installer
ok 7 run the installer
ok 8 run the installer once again
ok 9 wait 10 seconds
ok 10 check web app is up
ok 11 wake up puppet agent
ok 12 install all compute resources
ok 13 restart foreman
ok 14 install CLI (hammer)
ok 15 check smart proxy is registered
ok 16 check host is registered
ok 17 collect important logs
```

| | Number | |
|--|--------|---------------------------------|
| | 1 | |
| | 2 | |
| | 3 | |
| | 4 | |
| | 5 | configure repository |
| | 6 | install installer |
| | 7 | run the installer |
| | 8 | run the installer once again |
| | 9 | wait 10 seconds |
| | 10 | check web app is up |
| | 11 | wake up puppet agent |
| | 12 | install all compute resources |
| | 13 | restart foreman |
| | 14 | install CLI (hammer) |
| | 15 | check smart proxy is registered |
| | 16 | check host is registered |
| | 17 | collect important logs |

Package Testing - Automation

- Jenkins Slave
 - Clones foreman-bats git repo locally
 - Spins up VM on Rackspace using Vagrant
- Vagrant-rackspace
 - Shell provisioner clones BATS, syncs foreman-bats
 - Vagrant ssh from slave runs BATS, parses TAPS output
- Success triggers publishing of RPMs
 - (DEBs to come)

Out of Band Testing - Installer

- Puppet + wrapper library (kafo) + packaging
- Modules have individual tests (Travis & GitHub)
- Kafo has unit tests (Jenkins)
- Still need to verify the combined result
 - **Very** OS-dependant
 - Some free testing from the core build triggers
 - Manual: spin up VM, configure repo & clone BATS

Out Of Band Testing - Core

- Automated PR tests aren't always enough
 - Rake changes
 - Initializers
 - Packaging changes
- Packages can be built from:
 - Core PR number
 - Packaging branch
- Pushed to a staging repo for manual tests

Areas To Improve

- So far, so good:
 - Coverage is high
 - Developers can test arbitrary branches / PRs
- Would like to see:
 - DEBs tested before packages are published
 - More flexibility in testing changes to foreman-bats
 - Integration between BATS and scratch packages
 - More plugin tests & testing

Links

- Tools used
 - <http://jenkins-ci.org> / <http://travis-ci.org>
 - <http://www.vagrantup.com>
 - <https://github.com/sstephenson/bats>
- Foreman links
 - <http://theforeman.org>
 - <http://ci.theforeman.org>
 - <http://github.com/theforeman/foreman-bats>

Thanks!

- Questions?
- IRC: 'gwmngilfen' on
 - Freenode#theforeman
 - Freenode#theforeman-dev