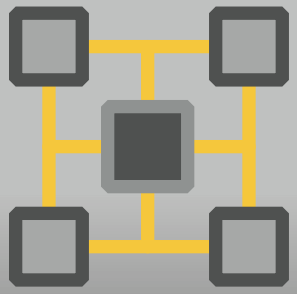


Research Computing

ARC Centre of Excellence for Particle Physics at the Terascale

Renewal of Puppet for Australia-ATLAS

Sean Crosby
Goncalo Borges
Lucien Boland
Jeremy Hack

Australia-ATLAS services

- ATLAS Tier 2
  - 100 WNs + Torque/Maui server
  - DPM headnode + 16 storage nodes
  - Regular EMI services (CE, BDII, APEL)
  - perfSonar
- CoEPP services
  - Public web servers, dokuwiki
  - LDAP/Kerberos auth servers
- Tier 3
  - UIs
  - NFS /home
  - CephFS /coepp/cephfs
- 250 nodes

History of configuration management

- Until 2013, cfengine2
  - Scripts written by previous admins
  - Covered SL5 services
- Lucien and I were not comfortable
  - Didn’t cover everything
  - Lots of hacks for bad packages or bugs fixed ages ago
  - Basically called YAIM for most Grid services
- But no overwhelming reason to change

Along came SL6 however…

Choices, choices…

- Quattor
  - Went to many EGI conferences, and was quite popular with French cloud
- Cfengine
  - Didn't really like syntax. Cfengine2 to cfengine3 was a big jump
- Puppet
  - Spoke with Steve Traylen at a HEPIX, and CERN was just getting into it
  - Lucien loves Ruby

Puppet basics

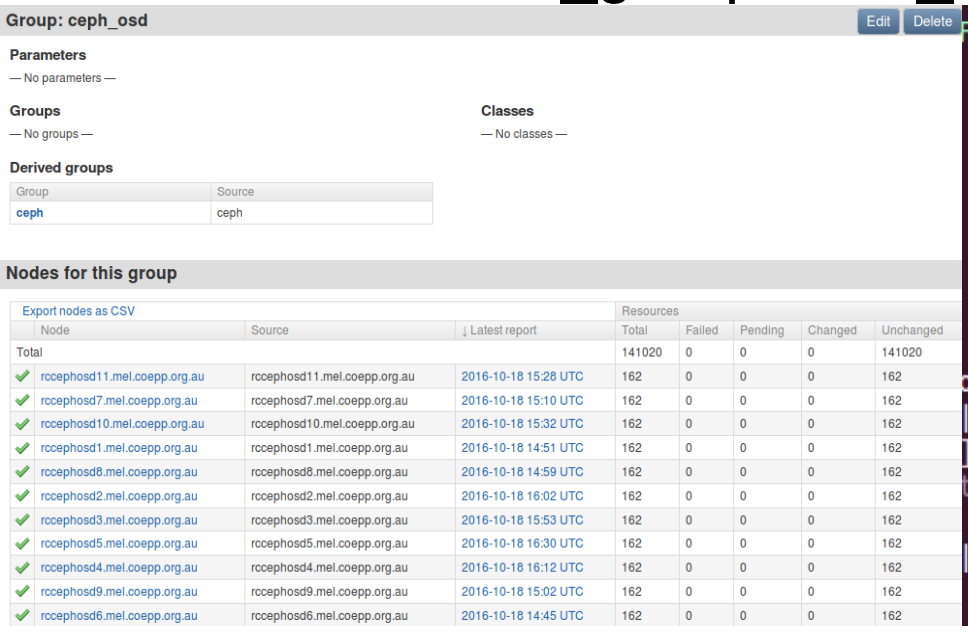
- Puppet is basically resources (file, service, package, exec, ssh_authorized_key,…) grouped into manifests and modules
- Puppet written in Ruby, and has its own DSL
- Facts are constants (OS version, IP address, hard drives etc) accessible to a Puppet run
- Hiera is a key/value lookup tool for configuration data, built to make Puppet better and let you set node-specific data without repeating yourself. (Puppet website)
- Forge is a Puppetlabs hosted community page where users can upload their own Puppet modules for others to use. CERN uploads most (all?) of their modules to here, as well as GitHub.

Original design

- Puppet 3.2/PuppetDB 1.5.2/mod_passenger/Puppet Dashboard
- All modules in single Git repository
- Hieradata in separate repository
- Environments were just branches of Git repo
- Git hooks

Original design

- Came from cfengine system of classifying nodes into groups (classes) for Nagios checks, common tasks
  - Puppet Dashboard hostgroups with custom Puppet parser function to pull contents from MySQL
  - Custom static Ruby Facter facts grouping hosts into host_group, host_type, location



Group: ceph_osd Edit Delete

Parameters

— No parameters —

Groups

— No groups —

Classes

— No classes —

Derived groups

Group	Source
ceph	ceph

Nodes for this group

Export nodes as CSV

	Node	Source	↓ Latest report	Resources Total	Failed	Pending	Changed	Unchanged
Total				141020	0	0	0	141020
✓	rccephosd11.mel.coepp.org.au	rccephosd11.mel.coepp.org.au	2016-10-18 15:28 UTC	162	0	0	0	162
✓	rccephosd7.mel.coepp.org.au	rccephosd7.mel.coepp.org.au	2016-10-18 15:10 UTC	162	0	0	0	162
✓	rccephosd10.mel.coepp.org.au	rccephosd10.mel.coepp.org.au	2016-10-18 15:32 UTC	162	0	0	0	162
✓	rccephosd1.mel.coepp.org.au	rccephosd1.mel.coepp.org.au	2016-10-18 14:51 UTC	162	0	0	0	162
✓	rccephosd8.mel.coepp.org.au	rccephosd8.mel.coepp.org.au	2016-10-18 14:59 UTC	162	0	0	0	162
✓	rccephosd2.mel.coepp.org.au	rccephosd2.mel.coepp.org.au	2016-10-18 16:02 UTC	162	0	0	0	162
✓	rccephosd3.mel.coepp.org.au	rccephosd3.mel.coepp.org.au	2016-10-18 15:53 UTC	162	0	0	0	162
✓	rccephosd5.mel.coepp.org.au	rccephosd5.mel.coepp.org.au	2016-10-18 16:30 UTC	162	0	0	0	162
✓	rccephosd4.mel.coepp.org.au	rccephosd4.mel.coepp.org.au	2016-10-18 16:12 UTC	162	0	0	0	162
✓	rccephosd9.mel.coepp.org.au	rccephosd9.mel.coepp.org.au	2016-10-18 15:02 UTC	162	0	0	0	162
✓	rccephosd6.mel.coepp.org.au	rccephosd6.mel.coepp.org.au	2016-10-18 14:45 UTC	162	0	0	0	162

```
Facter.add("host_group") do
  setcode do

    host_group = "unknown"

    # Get current ip address from Facter's own database
    hostname = Facter.value(:hostname)
    fqdn = Facter.value(:fqdn)

    if hostname.match("^agh4") || hostname.match("^agh3") || h
ostname.match(/^agc[\d]+/) || hostname.match(/^agcream[\d]+/)
|| hostname.match(/^agtorque[\d]?/) || hostname.match(/^ags[\d
]+/) || hostname.match(/^agtop[\d]+/) || hostname.match("^agsi
te") || hostname.match("^agargus") || hostname.match("^newse")
 || hostname.match("^agsitenew") || hostname.match("^rcsrm") |
| hostname.match("^rcgftp1")
      host_group = "tier2"
```

Original design

- Use Dashboard hostgroups to populate Nagios hostgroups
  - We did this due to slowness of exported resource compilation in Puppet 3.2

```
class nagios::server::hostgroups {
   $hostgroups = dashboardListGroups([])
   nagios::server::hostgroups::wrap{ $hostgroups: }
   nagios_hostgroup { "all":
      ensure  => present,
      members => "*",
      tag     => "rcmon_${environment}",
      alias   => "CoEPP RC Systems",
   }
}

define nagios::server::hostgroups::wrap() {
   $members      = join(dashboardHostsInGroup($name),",")
   $membergroups = join(dashboardGroupsInGroup($name),",")
   nagios_hostgroup { $name:
      ensure => present,
      tag    => "rcmon_${environment}",
   }
   if ($members != '') {
      Nagios_hostgroup[$name] { members => $members }
   }
   if ($membergroups != '') {
      Nagios_hostgroup[$name] { hostgroup_members => $membergroups }
   }
}
```

Original design

- Use Facter host_groups for Puppet manifests (e.g. iptables manifest)

```
<% if @host_group == "jet" and @host_type == "ramdisk" -%>
#--SSH - allow access from everywhere with rate limiting ----#
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -m limit --limit 10/min --limit-burst 10 -j ACCEPT
# Allow communication from meltorque
-A INPUT -s 192.231.127.52/32 -j ACCEPT
<% end -%>
```

- Use if statements for choices

```
class yum::package {
  # clumsy way of ensuring XenSevers get RHEL5 packages
  if $::operatingsystem == 'XenServer' {
    package { [
      'yum-priorities',
      'yum-protectbase', ]:
      ensure => present,
    }
  } else {
    if $::lsbmajdistrelease == '5' {
      package { [
        'yum-priorities',
        'yum-protectbase', ]:
        ensure => present,
      }
    }

    if $::lsbmajdistrelease == '6' {
      package { [
        'yum-plugin-security',
        'yum-plugin-protectbase',
        'yum-plugin-priorities', ]:
        ensure => present,
        require => Class['yum::repos::sl'],
      }
    }
```

Problems

- Single git repo made “safe” development harder
  - Changing module to support new version of software, but keeping existing clients running involved branching full git repo
    - Merging back was sometimes difficult due to many changes happening in main branch
- We wrote many modules before the Forge/CERNops was made
  - Harder to integrate 3rd party modules due to dependencies (e.g. DPM puppet modules, VOMS, MySQL)
  - Written in a “just get it done” way. Not very extensible or shareable

Problems

- Not every part of server was Puppeted
  - Some packages installed in Kickstart
  - Networking not configured (e.g. bonds, LACP, machines with static IP e.g. DHCP server)
  - perfSonar has a small Puppet config applied (ssh keys, firewall)
  - Xenservers also with a small Puppet config
  - At the start, still relied on YAIM for some Grid services
- Some machines not Puppeted at all
  - /home NFS server. We originally deemed it “too critical” to be Puppeted

Problems

- When a machine is not completely controlled by Puppet, it breeds a lack of confidence in server
  - We did manual config to get Puppet servers up and running to accept Puppet connections
  - Puppet servers haven't been updated in 3 years because we don't have complete confidence what was done to make them work
- Lots of steps can be missed when commissioning server
  - Forget to add host to Puppet Dashboard hostgroup stops monitoring
  - Forget to add host to Facter facts stop certain packages/iptables rules being added

Problems

- Harder to get other team members up to speed
  - “Why isn’t this host joining the right Ganglia cluster”
- Were not following best practice
  - Cool new features like auto Hiera lookup and structured data from Hiera were impossible for most of our modules without a complete rewrite

```
class {"infiniband::ib0": ib_addr => regsubst($ipaddress,'192.43.208','192.168.2','G')}
```

Impetus for change

- Moving virtualisation technology from Xenserver to KVM
  - Reinstall Puppetserver on KVM?
  - Could convert existing Puppetserver to KVM…
- Puppet 4 is coming
  - We relied on node inheritance – deprecated in Puppet 4
- Centos 7 servers
  - The final kick for us to move and rewrite

Puppet 4 changes

- Lots of deprecations
  - Node inheritance
  - “import” for manifests
  - Variables can’t start with capital letters
  - Class names can’t have hyphens
  - Updating array/hash values
  - Ruby DSL
  - Config file environments
  - Facts no longer stringified
- Cool new features
  - AIO packaging: newer version of Ruby – made exported resources so much faster
  - EPP templates

Many “best” practices

- We searched around, including in textbooks and websites, for “best” practice
- Settled on a few golden rules
  - Data is in Hiera
  - Always use Puppet variable autolookup
  - Default values for variables in module Hiera
  - Module name should reflect package name (with few exceptions)
  - Search Forge/CERNops first before writing (pick module with least dependencies)
  - 1 module per Git repo (r10k)
  - Roles/profiles
  - ENC sets all node intrinsic values
  - Puppet EVERYTHING, including Puppet

Puppet: A new way

- We created a set of bootstrap scripts, which just install puppet, run r10k, and enough config to then run puppet to install our puppet servers
- Implemented a separate CA server, to allow for easy scale up of Puppet servers
- Lots of our new modules are just copied/pasted from Forge

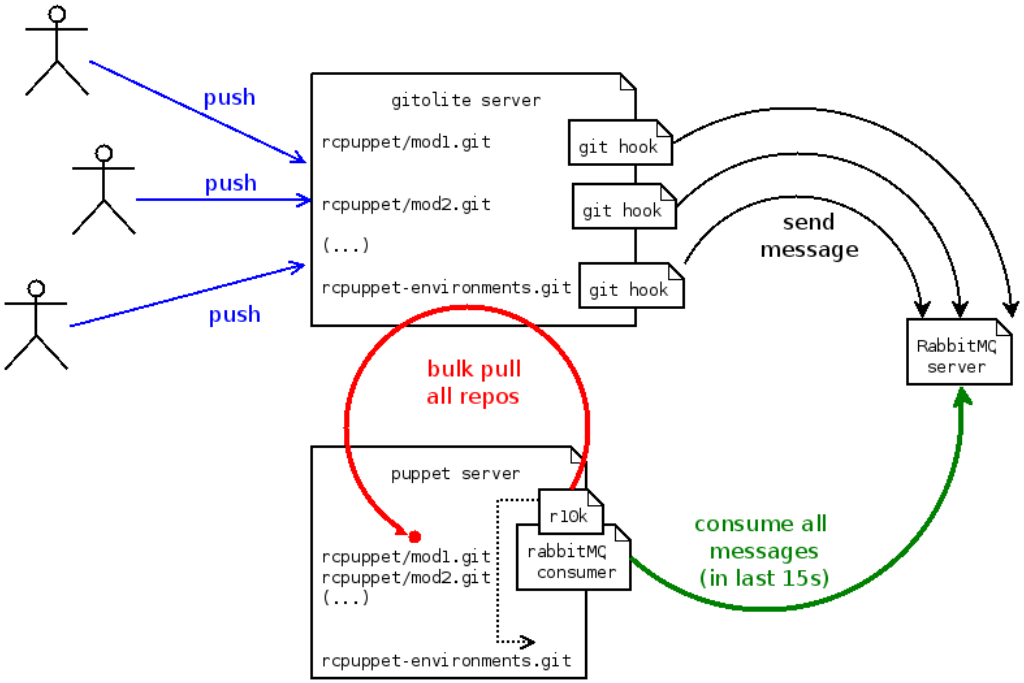
Puppet: A new way

- New, simple ENC
  - Hiera based
  - All “intrinsic” properties
  - Easy to add more if needed

```
04:18|scrosby@stewie: /data/git/rcgit.atlas/rcpuppet/rcpuppet-environments (production)$ cat encdata/certs/melvpn.mel
.coepp.org.au.yaml
---
classes:
  roles::%{hiera('parameters.enc_role')}:
parameters:
  enc_role: 'melvpn'
  enc_location: 'melbourne'
  enc_hardware: 'dell_r710'
  enc_functional_group: 'infrastructure'
environment: production
```

Puppet: A new way

- Better and simpler Git hooks



Puppet: A new way

- Simpler way to customise modules based on Facts

```
17:14|scrosby@stewie: /data/git/rcgit.atlas/rcpuppet/rcpuppet-environments (production)$ cat hieradata/coepp/os/CentOS-7.yaml
---
packages::install: ['bash-completion','bash-completion-extras','dmidecode','git','htop','iptraf','nc','openssl','ruby','ruby-irb','ruby-doc','scl-utils','strace','tcpdump','tmux','vim-enhanced']
```

- Easier to customise/disable Nagios checks for hosts
  - Old hostgroup model made it very difficult to disable a check for a specific host
- Module writing is actually easier
  - No if statements, no edge cases

Summary

- Old Puppet worked, but was not optimal
- Centos 7 was last push for us to change
- “Best practices” are best for a reason
- New system is easier to commission a host, and easier to maintain
- We love the Forge – please share your modules!