Puppet at USCMS-T1 - Year 2

Tim Skirvin
USCMS-T1 @ FNAL
tskirvin@fnal.gov

Supported in part by the Department of Energy
DE-AC02-07CH11359

A Few Things About Puppet

- I'm not going to describe the tool in detail.
- It's not the only solution out there, but it's a pretty good one, and it seems to be winning in HEP.
  - Up-and-coming tools: ansible, salt
- Has the strengths and weaknesses of a programming language (but it isn't one).
- Puppet and Ruby are both moving targets.
  - Collaboration with the rest of the world depends on following then-current shared best practices
- Puppet Labs is still facing the startup "Market Share vs Income" question.
  - They're pushing Puppet Enterprise; we don't want it.

Previously…

- …we wanted our entire team to be able to use Puppet.
- …we wanted to scale to ~1500 nodes and retire ROCKS + SLF5 entirely.
- …we wanted to separate data vs code …we wanted to collaborate with external organizations + FermiLab.
- …we wanted more and better internal documentation.
- …we wanted to separate pets vs cattle.
- …we wanted to better use test suites.

http://indico.cern.ch/event/247864/session/6/contribution/44

How have we responded to these challenges?

- Puppet has become our primary sysadmin tool.
- We have scaled to ~1250 nodes (~70 hosts to convert, ~200 hosts on the way).
- Code separation has been supported by:
  - hiera + roles/profiles: data-vs-code separation work
  - r10k: project will allow us to improve work on modules
- We collaborate via local puppet-users list, HEPiX Configuration Management WG, and OSG puppet group.
- Internal documentation has been improved.
- We're still coming to grips with the 'pets-vs-cattle' argument.
- We're still behind on test suites.

Pets vs Cattle

- Ideally, we want our hosts to be interchangeable (cattle) instead of carefully tended (pets)
- Interchangability is at least partially a social issue
  - Does it require a special piece of hardware? It's a pet.
  - Does it need a specific IP or hostname? It's a pet, unless:
    - You can use a separate service IP
    - You can load balance the whole thing.
  - Can you get by with 'cmssrvXXX' hostnames instead of 'cms-puppet3'?
    - How about CNAMEs?
    - Can your networking department act fast enough?
- We still haven't worked this out to our satisfaction.

What has changed?

- Current version: puppet 3.6.2 (we're behind a bit)
- hiera has won the day
- Directory Environments
- r10k for managing external modules
- Roles + Profiles have mostly won
  - …but they remain oddly undocumented
- Puppet Dashboard is dead
  - Replacement: https://github.com/nedap/puppetboard
- Puppet 4 is coming within ~6 months
  - Will have a new parser
  - Lots of deprecated code will be entirely removed

Puppet Philosophy @ USCMS-T1

- Everything should be in puppet.
- Everything lives in git, even the things that aren't in puppet.
- It should be easy to tell what a host is doing.
- Code goes into modules; data goes into hiera
- We should be able to reinstall any host and get its service up-and-running automatically.
- Code should be reviewed by at least one other person before it goes into the main repo.
- We need to balance the needs of code re-use and service supportability
- We should be able to share our work.

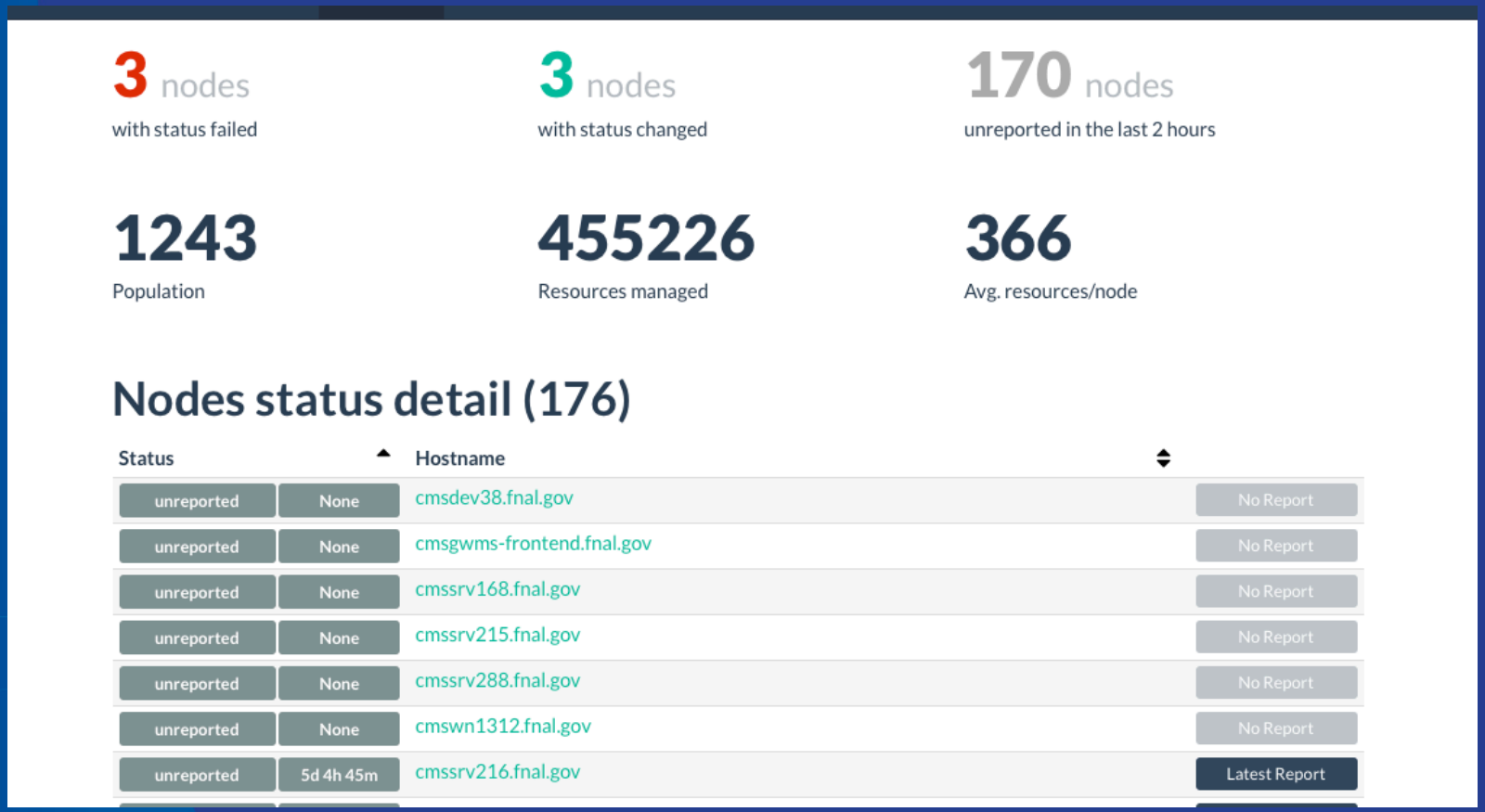
New version of puppet (3.6.2)

- Not quite bleeding edge (but close)
- Upgraded from 3.3.x branch:
  - Executes code in the order it appears in the code -> improved debugging
  - Performance improvements for server + puppetdb
  - Supports Directory Environments
- It was fairly painless
  - Upgrade facter at the same time!
- Will probably be our last major upgrade before Puppet 4
  - This one will be a Project.

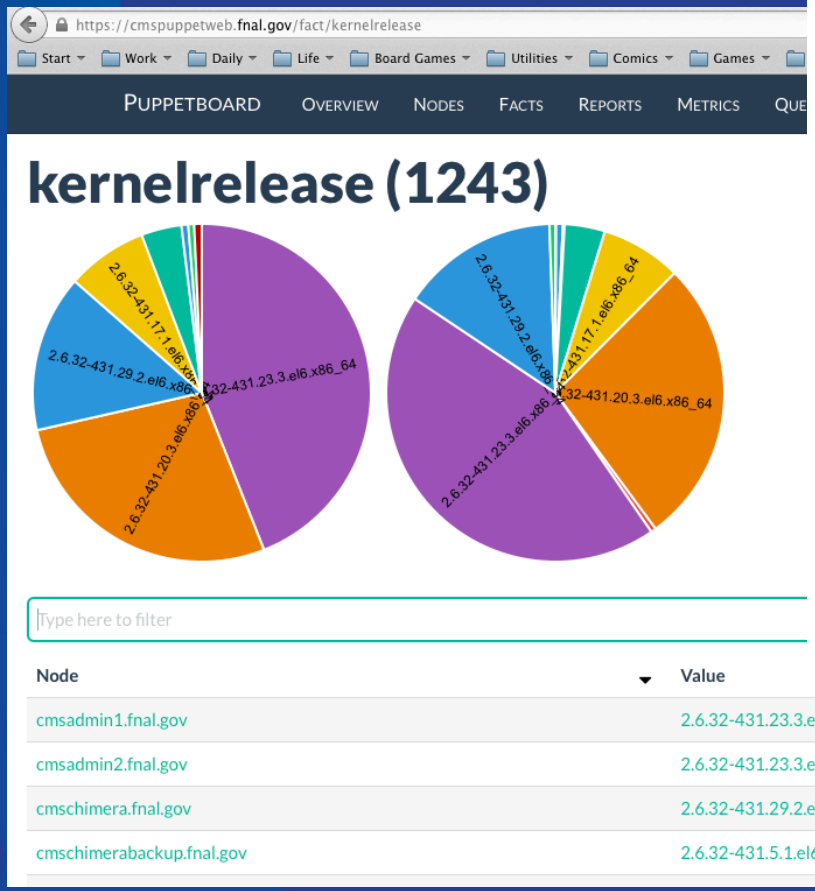
puppetdb integration

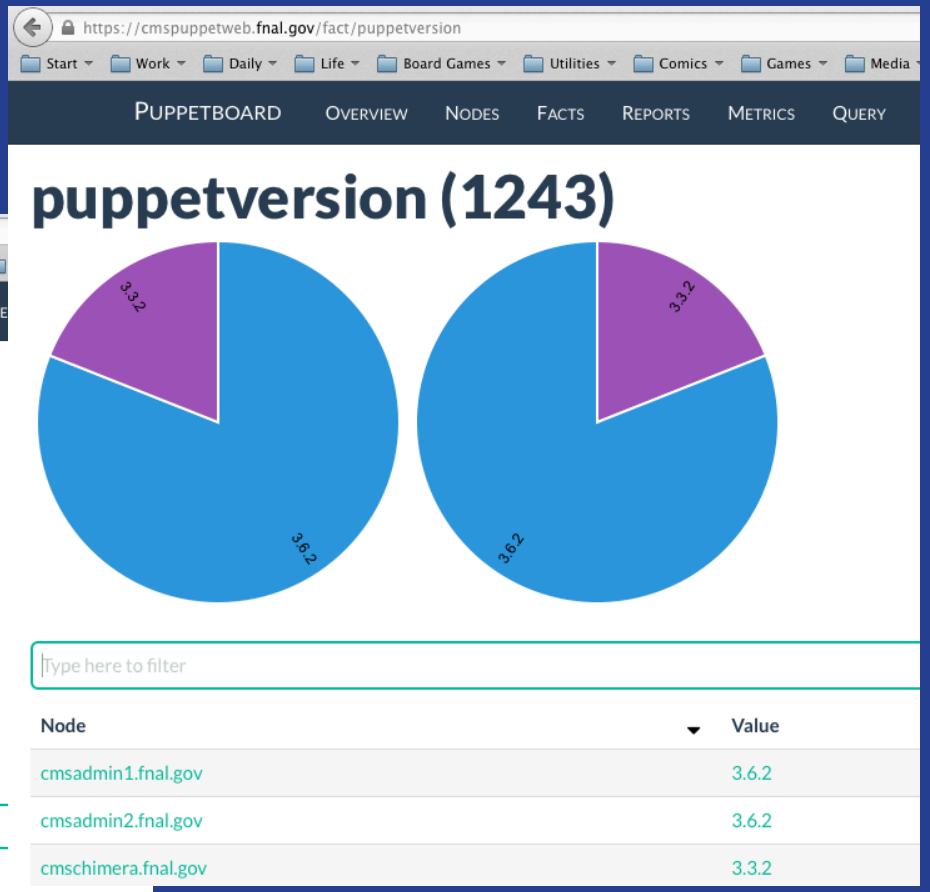
- https://github.com/tskirvin/cms-puppetdb-tools
- We mail weekly reports before our group meeting
  - "Too quiet" hosts – lets us spot down or dev hosts
  - "Failed" – usually finds problems in development
  - "Tangled" – shows us hosts that are doing the same thing again and again. Finds development problems.
- https://github.com/nedap/puppetboard
  - Web interface that is just a client to the puppetdb
  - Very shiny
  - Uses very few local resources
  - https://github.com/nedap/puppetboard#screenshots

Puppetboard – Screen Shot (Top Level)



Puppetboard – Screen Shots II





Dynamic Directory Environments

- Git environments are now fully integrated with Puppet.
- Replaces ‘config-file environments’
  - Old: `modulepath=$confdir/environments/$environment/modules`
  - New: `environmentpath=$confdir/environments`
- `environment.conf` now defines `modulepath` (and similar)
- WARNING: This is an all-or-nothing change!
  - `basemodulepath` looks like it would allow a seamless transition; this is a trap.
  - In puppet 3.6.x, $environment parses as your default environment (e.g. ‘production’)
  - In puppet 3.7.x, $environment throws an error
  - Bug reports have been filed.

https://docs.puppetlabs.com/puppet/latest/reference/environments.html

Role + Profiles – becoming first-class citizens

- Puppet Labs endorses the Role/Profile mix now
- Each host has one-and-only-one role (set by the ENC)
- Each role has a number of associated profiles (e.g. ‘web server’ or ‘puppet client’)
- We use a separate ‘profiles/*’ directory.
  - This is apparently novel? Most sites use modules/profile?
  - Profiles are named ‘p_*’ to avoid name conflicts
  - Helps provide an “upgrade path” from profiles to full-blown modules.

r10k

- https://github.com/adrienthebo/r10k
- Provides a convenient way to pull in modules from external resources
  - …and to share your code with collaborators.
- You don't have to replace your 'modules' directory.
- Works nicely with Directory Environments.

```
PUPPETFILE=./Puppetfile \
    PUPPETFILE_DIR=r10k-modules \
    /usr/bin/r10k puppetfile install
```

External Node Classifier (ENC)

- We wrote our own.
  - Each host has its own .yaml file (in git)
  - Each file defines the system role, associated flags, and (if non-standard) the environment
  - There is a ‘default’ file for hosts without a .yaml file
- Limitations
  - The environment in the ENC is canonical, you can’t even reset it from the command line (makes testing harder)
- Puppet Labs claims to be ready to release a default ENC in the near future

/etc/facter/facts.d

- With puppetlabs-stdlib, you can set facts by dropping files into /etc/facter/facts.d
  - Example: role.txt reads ‘role=console’ on cmsconsole
  - Example: manager.txt reads ‘manager=tskirvin’ on cmsdev43
- You can put scripts in here too.
- Provides a nice way to let hosts discover their own world in puppet
  - ‘facter role’ can be run to find out what the host does
  - Can put this data into the MOTD
- Provides a good hook for puppetdb searches

Monitoring Puppet

- Zabbix checks to make sure that puppet agent is running/succeeding on all hosts
  - /var/lib/puppet/state
- We use puppetdb to track data in aggregate
- We use ‘puppet agent –disable’ to stop puppet rather than just stopping the process
  - Makes zabbix happier

Puppet + Git

- Everybody writes code in their own branch
- Code is promoted to testing (‘itb’) branch based on the okay of one colleague
- Code is promoted from itb -> production based on the okay of the whole team
  - Generally done weekly, after the group meeting
- Emergency changes can go straight to production, but are extremely rare
- There is also a “data-only” code branch
  - Example: users/groups, DNS, htpasswd files
  - Doesn’t go through the same process

Continuous Integration

- We're still not doing it.
- The future (currently) appears to be in Beaker.
  - Puppet Labs product
  - Doesn't currently work out-of-the-box RHEL6/SLF6 systems (bugs have been filed)

PuppetConf 2014 Report

- Next year will be in Portland (was San Francisco)
- Moving from a monolithic model to ‘multiple apps’
  - Can focus on each tool individually
  - Examples: puppetdb, puppet server, an ENC
- The relationship between Puppet Enterprise and Puppet Open Source is still not clear
  - They’re not abandoning Open Source, but they still need to make money
- Puppet Approved modules are coming (depending more on Continuous Integration requirements)
- The Technology of the Year is Docker.
  - …but I’m not going to talk about it. Now, anyway.

hiera-eyaml

- https://github.com/TomPoulton/hiera-eyaml
- Secret management within .yaml hiera files
  - Easier to track secrets and changes to secrets.
  - Faster because it's not GPG.
  - Still just as vulnerable as before (decrypted passwords appear in the puppetdb and in system logs).
- We haven't started using it yet.
- Keeper project is about ready for deployment
  - Will be used to deploy non-password secrets, i.e. keytabs
  - Deploys files via ssh

Puppet + Monitoring + Service Now

- Ideally, puppet should be the “source of truth” for monitoring systems
  - We’ve got a ways to go here.
- We are using Nagios + check_mk to monitor “back-end” systems – disk boxes, PDUs, console servers
  - check_mk gets host lists via puppet
  - Infrastructure is there to auto-populate host lists on an arbitrary basis, we just have to commit
- Automatically creates incidents in Service Now
  - Closes them out again when the incident goes away
  - https://github.com/tskirvin/fnal-nagios
  - https://github.com/tskirvin/fnal-snow

What comes next?

- Retire local modules that should be in r10k
- Finish the ROCKS/SLF5 retirement
- Try again to formalize the pets/cattle division
- Keeper
- Do we keep cobbler?
- Puppet 4
- Maybe add another puppet server?
- Continuous Integration/testing?
- How best to use ‘system manager’ facts?
- How much do we need Puppet Enterprise?