Configuration Management @CERN

From homegrown to industry standard
agenda

• Before Puppet: a brief history of systems management at CERN

• Current Puppet infrastructure

• Future plans, what works, things we like, things we don’t quite get yet

• Questions
CERN background

- Currently < 12K nodes
- Modest size for compute
- Data is another story...
- Analysis geographically distributed
config diversity

- Lots of applications “clusters”, lots of admins
quattor

- “Extremely Large Fabric Management System”
  - http://cern.ch/ELFms
- EU DataGrid project
- Toolchain includes “Quattor” for configuration management
- At time of project - 2002, less choice in config management.
quattor from 30K feet

- **Declarative template language**
  - 
    ```
    "system/components/useraccess/allow" = list('bejones', 'mccance', 'straylen');
    ```

- **Templates compiled into machine profiles**
  - xml or json
  - schema checking

- **Machines notified of changes**
  - fetch profile if newer

- **Software components on machine configure from profile**
  - register namespace
  - only run on changes
quattor problems

- Templates describe entire plant
  - some benefits, such as client/server mappings
- Waiting for compiles
  - spaghetti dependencies mean multiple profiles compiled for simple changes
  - easy to break everything
- No “facts” - you have to tell the target everything about what it is.
- Sweet spot is lots of commonality (ie large clusters)
Renovation

The Agile Infrastructure

Making IT operations better since 2013

Puppet

Openstack

Foreman

Hiera

mcollective

Koji

git

ActiveMQ

Ben Jones - Puppetconf 2012
hardware provisioning

• Homegrown tool in use to bootstrap
  – Add to other stores (network db, foreman)
  – Burn-in, DOA etc
  – Final step of tool adds host to foreman

• We are very happy with foreman
  – Kickstart templating is great
  – Hostgroup organization is analogous to our old quattor clusters
  – Looking to use for OpenStack integration
  – API!

• This solution is similar to Razor
  – tracking Razor at the moment
virtual provisioning

- Pre-existing infrastructure Microsoft HyperV
  - Pre-register in foreman
  - Kickstart installations to puppet & foreman
- Puppet managed OpenStack Nova
  - 1000 VM boinc SixTrack (LHC@home)
  - 4000 VM batch test bed
  - Aim to support <20K hypervisors with density up to 20:1
  - Machine images via Oz
  - No pre-registration in foreman
  - amiconfig & cloudinit for contextualization
scale out
change process

- Git service used for puppet modules & manifests
- Git branches map to dynamic environments
  - admins push changes to a (gitolite) repository
  - puppet masters pull branches and translate to environments
  - Production, Testing & Devel branches
  - Topic branches for major changes
  - Some services live in their own branches
    - risk of divergence...
- Atlassian Crucible & Fisheye for module review process
sharing modules

• We aim to share modules as much as possible.
• Want to be a good citizen, but also other related deployments
  – CERN IT not only puppet deployment onsite
    • ATLAS point 1 farm
  – ATLAS analysis in the cloud
  – International HEP labs using or moving to puppet
• http://github.com/cernops
puppet modules

• Initial hope: download, install, forget about it!
  – At least all common components done? mysql, sshd, sysctl, apache etc
  – holistic frameworks

• Reality a little messier
  – too simple: Package/File/Service
  – more variation than, eg, different location of ssh config for each OS.
  – too complex!

• Our own initial modules littered with CERNisms
  – services (ie ntp servers), auth systems, subnets
separate code & data

• One of the things quattor did well that we miss
• ENC globals bad for module reuse
  – people who are used to foreman like using global parameters
• hiera the answer
  – hiera yaml files distributed with modules & manifests
  – hiera gpg for secrets (replacing another CERNism...)
  – DB backend for integration with other systems (ie monitoring metrics)
:hierarchy:
  - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}/%{hostgroup_2}/%{hostgroup_3}/%{hostgroup_4}
    - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}/%{hostgroup_2}/%{hostgroup_3}
      - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}/%{hostgroup_2}
        - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}
    - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}/%{hostgroup_3}
  - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}/%{hostgroup_2}
  - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}/%{hostgroup_1}
  - environments/%{environment}/hieradata/hostgroups/%{hostgroup_0}
  - environments/%{environment}/hieradata/environments/%{environment}
  - environments/%{environment}/hieradata/module_names/%{module_name}
  - environments/%{environment}/hieradata/common
  - hieradata/hostgroups/%{hostgroup}
  - hieradata/common
hiera & hostgroups

• With our configuration diversity useful to nest data
• Map to foreman hostgroup hierarchy
community module support

- Fun to work with a vibrant community
- Problems are getting fixed for us before we have chance to submit issues.
- Openstack
  - started with our own basic modules
  - initial community modules ubuntu based
  - support for first fedora then rhel-alikes
  - looking forward to helping with future improvements for complex configurations
open questions

• Puppet is better at solving our diversity, still problems with multiple admins
  – Core team vs service managers
  – Need to avoid having a dozen ways to configure dns
• Rolling updates
• Multiple entry points to install, who is “correct”
  – nova images vs kickstart
  – foreman to control OpenStack
• Test workflow with hiera
future plans

• PuppetDB
• mcollective
  – set host status: ie “draining”, “maintenance”

```
[root@tpsrv680 agent]# cat /usr/libexec/mcollective/mcollective/agent/hoststate.rb
module MCollective
  module Agent
    ...
    action "set" do
      validate :msg, String
      reply[:msg] = request[:msg]
      reply[:status] = run("echo host_status=#{request[:msg]} > /etc/facter/facts.d/host_status.txt;date >> /var/log/host_status.stack; echo '#{request[:msg]} >> /var/log/host_status.stack", :stdout => :out, :stderr => :err, :cwd => "/tmp")
    ...
  ...
```

  – more use as plant expands

• Some people don’t want to learn puppet!
  – cater for applying simple recipes
questions?
Architecture diagram
LAST NIGHT I DRIFTED OFF WHILE READING A LISP BOOK.

SUDDENLY, I WAS BATHED IN A SUFFUSION OF BLUE.

Huh?

AT ONCE, JUST LIKE THEY SAID, I FELT A GREAT ENLIGHTENMENT. I SAW THE NAKED STRUCTURE OF LISP CODE UNFOLD BEFORE ME.

MY GOD

IT’S FULL OF CAR’S

THE PATTERNS AND METAPATTERNS DANCED.

SYNTAX FADED, AND I SWAM IN THE PURITY OF QUANTIFIED CONCEPTION. OF IDEAS MANIFEST.

Truly, this was the language from which the gods wrought the universe.

NO, IT’S NOT.

IT’S NOT?

I MEAN, OSTensibly, YES. HONESTLY, WE HACKED MOST OF IT TOGETHER WITH PERL.