Fermilab

U.S. DEPARTMENT OF ENERGY | Office of Science

Introducing the new GCSO puppet service

Gerard Bernabeu

GCS Dept. Meeting

March 5th 2014

Index

- Puppet 101
- Puppet roles/profiles design pattern
- Puppet librarian/R10k
- GCSO puppet server architecture

You can find all of this information in GCSO Puppet wiki

Puppet 101

What is puppet?

- Think of it as a language.
- Describe state, not steps.
- Paint a picture of your *ideal* and *most clean* system.
- Puppet does the rest!

Why do I want it?

- You can do cool things with it. Human-parsable.
- E.g.: ensure that all your hosts are running ssh
  - **service {'ssh': ensure => running}**
- Learn puppet?
  - http://docs.puppetlabs.com/learning/
  - http://docs.puppetlabs.com/puppet_core_types_cheatsheet.pdf

Puppet Roles / Profiles Design Pattern

This is a widely used design pattern for puppet code (classes and node definitions).

Puppet roles and profiles design pattern is properly explained & documented here:

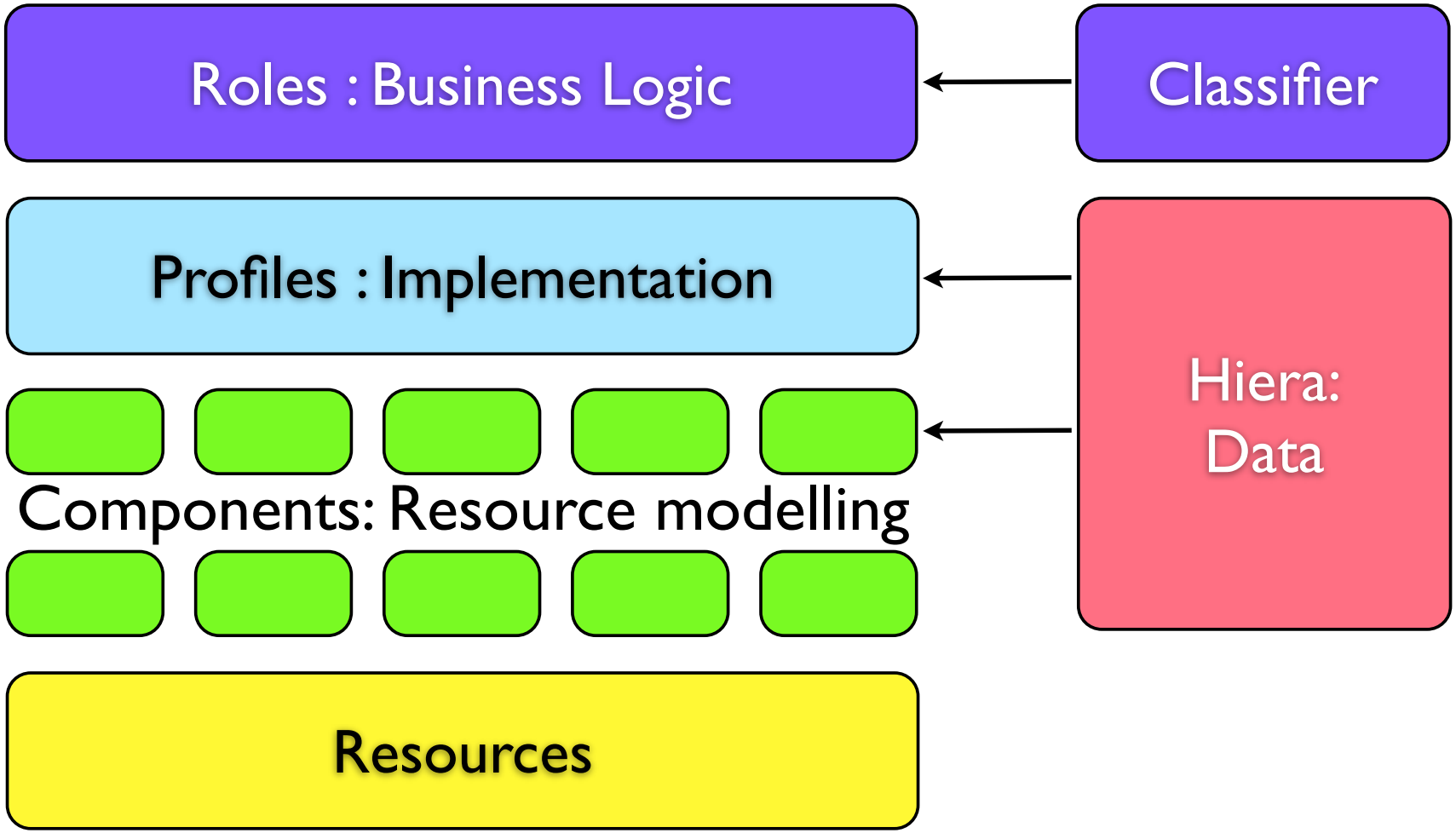
- http://www.craigdunn.org/2012/05/239/
- http://www.slideshare.net/PuppetLabs/roles-talk
  - Also very good at showing how different groups can collaborate.

Puppet Roles / Profiles Design Pattern

Danger Signs

- Resources being declared in two modules
- You don't know where your implementation "fits"
- Lot's of logic at a node level
- Repetition and duplication
- The *if* statement is your go-to-guy

The Stack


Roles : Business Logic
Classifier
Profiles : Implementation
Hiera:
Data
Components: Resource modelling
Resources

Puppet Roles / Profiles Design Pattern

- Profiles and Roles *are Puppet modules.*
- Components are Puppet modules responsible for modelling resources.
- Everything is a module.
- Components should be named after what they manage (apache, ssh, mysql)
- Profiles should be named after the logical stack they implement (database, bastion, email)
- Roles should be named in business logic convention (uat_server, web_cluster, application, archive)

Puppet is all about abstraction

- Data is abstracted by Hiera
- Providers are abstracted by types
- Resources are abstracted by classes
- Classes are abstracted by modules
- **Modules are abstracted by profiles**

Puppet Roles / Profiles Design Pattern

The fundamental concepts:

- Abstraction, abstraction, abstraction.
- Decoupling business logic, implementation and resource modelling.
- Separating data and code.
- Reducing node-level complexity.

- Use Hiera to model your data.
- Use profiles to model your implementation.

Puppet Roles / Profiles Design Pattern

A GCSO specific example: FermiCloud hypervisor

```
node 'fcl005.fnal.gov' {
  include role::opennebula::hypervisor::rhcluster
}
class role::opennebula::hypervisor::rhcluster inherits role::opennebula::hypervisor {
  include gfs2
  include cluster
}
class role::opennebula::hypervisor inherits role {
  include '::libvirt'
  include '::one::gratia-probe'
  include '::one::hypervisor'
}
class role {
  include 'profile::puppet::agent'
  include 'profile::gcsobase'
}
class profile::puppet::agent {
  include '::puppet::agent'
  include '::puppet::checkmk'
}
```

Puppet Roles / Profiles Design Pattern

A GCSO specific example: FermiCloud hypervisor - HIERA

```
$ cat hieradata/hostgroup/fcl.yaml
#any system named fcl* will fit in here. They should only be FermiCloud baremetals
libvirt::checkmk::allregistered: false

$ cat hieradata/defaults.yaml
#Puppetmaster with puppetdb and CA authority can only be a single server, gcsopuppet.fnal.gov for GCSO production env
puppet::master::storeconfigs_dbserver: 'gcsopuppet-puppetdb.fnal.gov'
puppet::master::manifest: '/etc/puppetgcso/environments/$environment/site.pp'
puppet::master::hiera_file: 'puppet:///modules/profile/puppet/hiera.yaml'
puppet::master::ca_server: 'gcsopuppet-ca.fnal.gov'

#The puppet_server is a RR alias in order to distribute the load
puppet::agent::puppet_server: 'gcsopuppet-rr.fnal.gov'
puppet::agent::ca_server: 'gcsopuppet-ca.fnal.gov'
puppet::agent::environment: %{::environment}
puppet::agent::puppet_run_style: 'cron'
puppet::agent::puppet_rundir: '/var/run/puppetgcso'
puppet::agent::puppet_vardir: '/var/lib/puppetgcso'
puppet::agent::puppet_logdir: '/var/log/puppetgcso'
```


Fermilab

Puppet librarian / R10K

R10K provides a general purpose toolset for deploying Puppet environments and modules. It implements the Puppetfile format and provides a native implementation of Puppet dynamic environments. https://github.com/adrienthebo/r10k

A Puppet environment is created for each GIT branch in the 'puppetrepo' GIT repository. This is the r10k setup:

```
# cat /etc/r10k.yaml
:cachedir: '/var/cache/r10k'
:sources:
  :local:
    remote: 'ssh://p-puppetrepo@cdcvs.fnal.gov/cvs/projects/puppetrepo'
    basedir: '/etc/puppetgcso/environments'
# This directory will be purged of any directory that doesn't map to a
# git branch
:purgedirs:
  - '/etc/puppetgcso/environments'
```

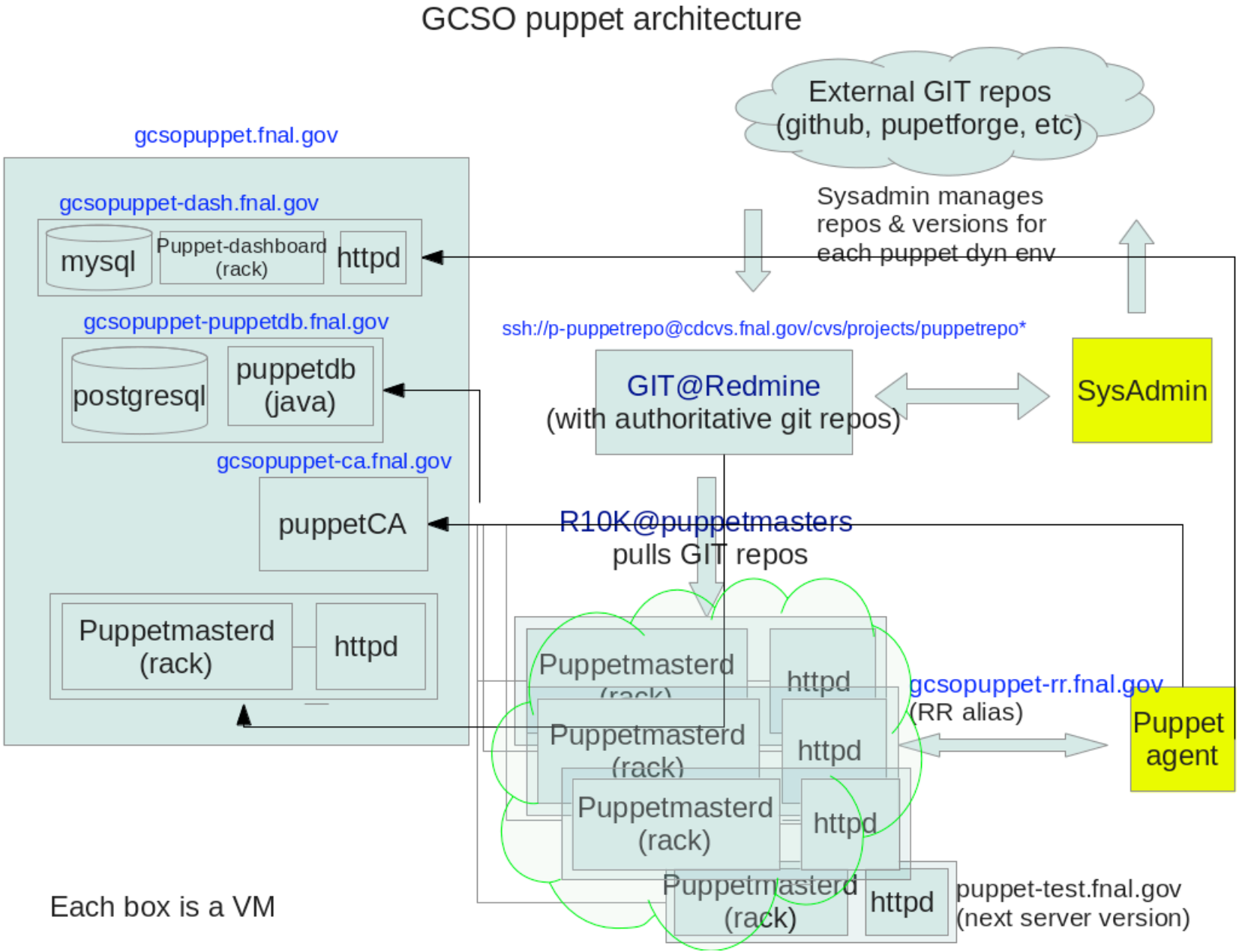
Puppet R10K Puppetfile

```
forge "http://forge.puppetlabs.com"

#Basig modules
mod "puppetlabs/stdlib", "4.1.0"
mod "puppetlabs/inifile", "1.0.1"
mod "puppetlabs/concat", "1.0.1"
#Management of cron.d entries (FEF has requested GCSO to transition from crontab to cron.d files)
mod "torrancew/cron", "0.1.0"

#Services we want to provision with puppet (some are requirements too)
mod "puppetlabs/apache", "0.11.0"
mod "puppetlabs/mysql", "2.1.0"
mod "puppetlabs/postgresql", "3.3.0"

#Puppet service to deploy puppetmasters, it comes with a bunch of dependencies.
mod "puppetlabs/puppetdb", "3.0.1"
mod "puppet",
  :git => 'ssh://p-puppetrepo@cdcvs.fnal.gov/cvs/projects/puppetrepo-stephenrjohnson-puppet'

#Modules we need for FermiCloud
mod "libvirt",
  :git => 'ssh://p-puppetrepo@cdcvs.fnal.gov/cvs/projects/puppetrepo-libvirt'
mod "cluster",
  :git => 'ssh://p-puppetrepo@cdcvs.fnal.gov/cvs/projects/puppetrepo-cluster'
mod "gfs2",
  :git => 'ssh://p-puppetrepo@cdcvs.fnal.gov/cvs/projects/puppetrepo-fermi-gfs2'
mod "one",
  :git => 'ssh://p-puppetrepo@cdcvs.fnal.gov/cvs/projects/puppetrepo-one'
```

Puppet R10K – directory tree

```
production]# tree -L 2
.
├── hieradata
│   ├── certname
│   ├── defaults.yaml
│   ├── environment
│   └── hostgroup
├── local
│   ├── mainserver.pp
│   ├── profile
│   ├── puppetfef
│   ├── r10k
│   ├── README.txt
│   └── role
├── modules
│   ├── apache
│   ├── cluster
│   ├── concat
│   ├── cron
│   ├── gfs2
│   ├── inifile
│   ├── libvirt
│   ├── mysql
│   ├── one
│   ├── postgresql
│   ├── puppet
│   ├── puppetdb
│   └── stdlib
├── nodes.pp
├── Puppetfile
└── site.pp
```

GCSO puppet server architecture



Each box is a VM

GCSO puppet server architecture

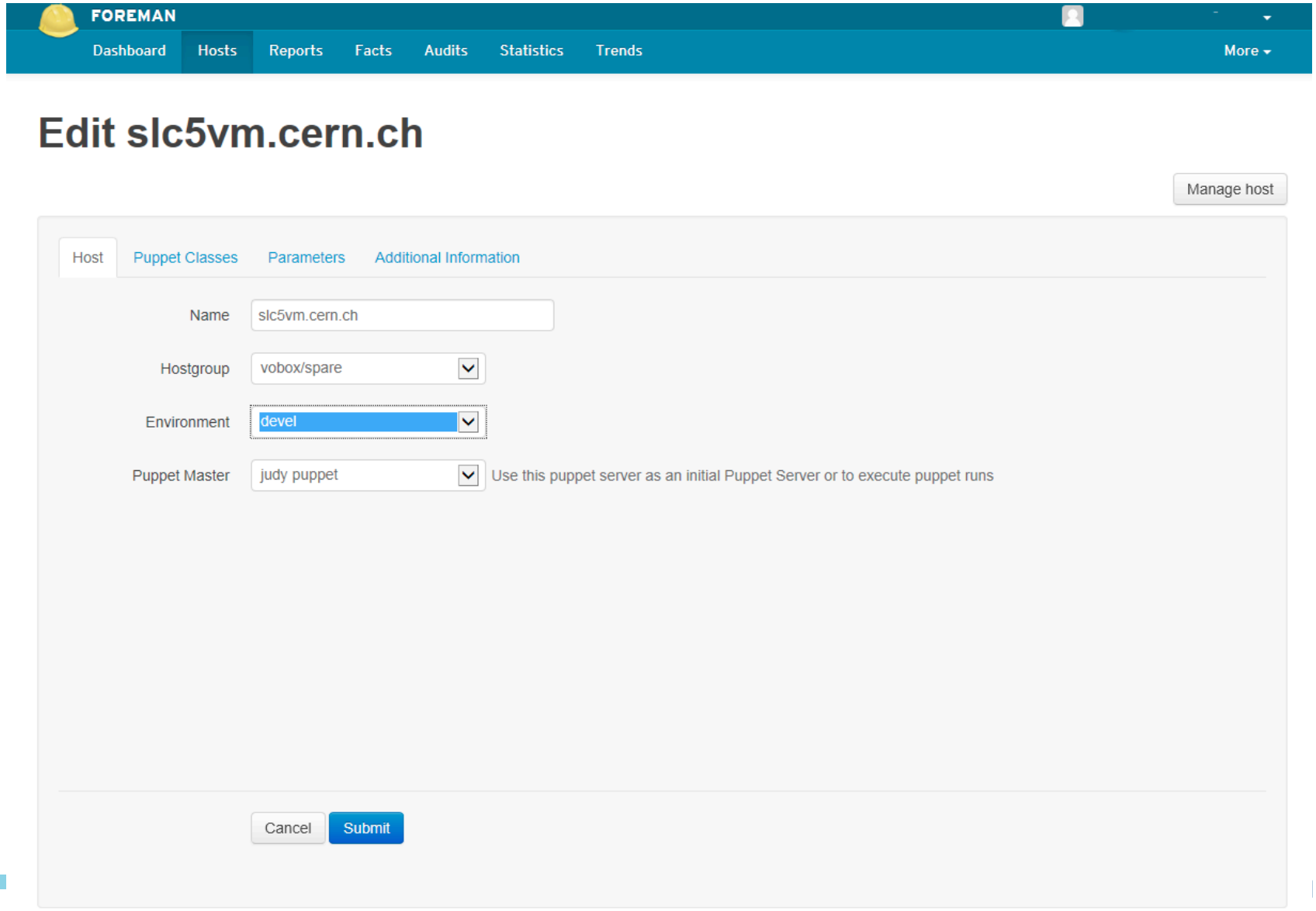
- All data GIT repos stored @ redmine
  - https://cdcvs.fnal.gov/redmine/projects/puppetrepo/repository

- Puppet users can create their own out-of-band puppetmaster
  - Rapid puppet code writing/testing
  - Totally out-of-band if using custom environment (agents are setup to stick to the last environment they run)

- With a properly setup External Node Classifier (ENC, eg: Foreman) it is possible to configure a node with a click of a button (assigning a preexisting role).

https://cdcvs.fnal.gov/redmine/projects/grid_and_cloud_computing_operations/wiki/GCSO_puppet_architecture

What may come next? – External Node Classifier (ENC)



ilab