

# Screw the Golden Master

## Configuration Management with Puppet





# Gary Larizza



- Director of Technology for Huron City Schools in Ohio
- Puppet Environment of approximately 400 nodes
- Using Puppet since Spring of 2009



# Agenda

- ✦ The problem with images
- ✦ What is Puppet?
- ✦ Why do we use it?
- ✦ Demo
- ✦ Questions



# Swag?

- ✧ PuppetLabs has sent some cool swag!
- ✧ It's not here yet.





# The Old Way





# A Gold Master for Everything

- ✦ Model Image
- ✦ Netrestore
- ✦ Wash, Rinse, Repeat





# State Drift

- ✦ Updates
- ✦ 'New' Software for Install
- ✦ Immediate Changes
- ✦ One-off modifications
- ✦ Image sprawl





# In Summary:

- ✦ Too many Images/Configurations
- ✦ Duplicated efforts
- ✦ MCX isn't comprehensive
- ✦ Don't need to manage EVERYTHING
- ✦ Screw that note/memo crap
- ✦ I'm Cheap



“I say never be complete.”





# What is Puppet?

- ✦ Configuration Management
- ✦ Ruby DSL - still very 'Ruby-like'
- ✦ Configured State vs. Running State
- ✦ Choose your level of commitment
- ✦ Useful with Local Admins
- ✦ Centralized management

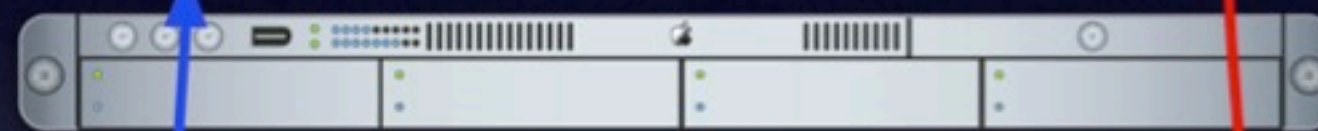
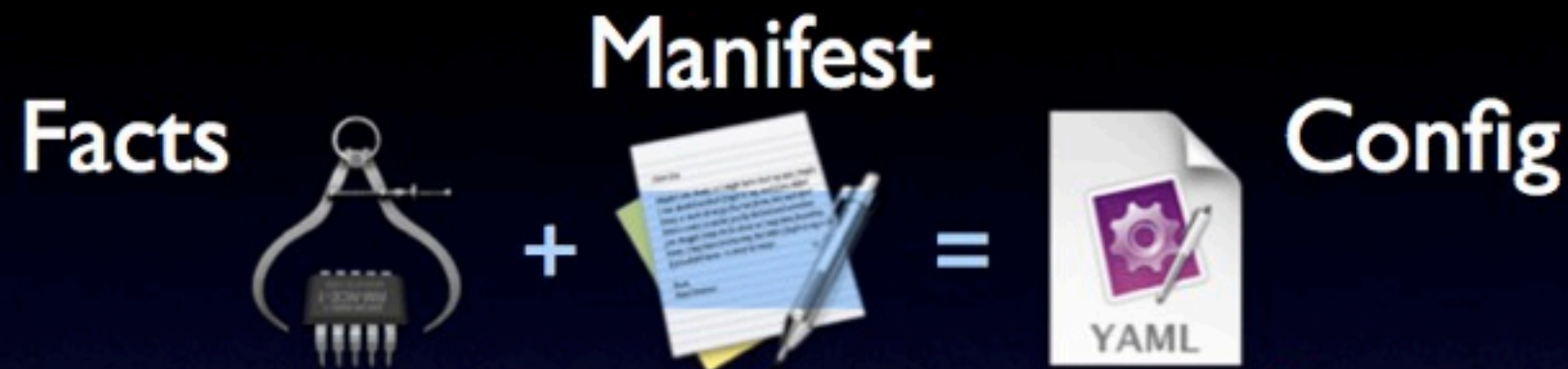


# Facter

- ✦ Describes aspects of your machine - “facts”
- ✦ Facts written in Ruby...for now
- ✦ Nice library of existing facts
- ✦ Custom facts are easy
- ✦ End up ‘shelling out’



# Dynamic Configuration



puppetmasterd  
server

The client collects  
and sends *Facts*  
about the  
workstation  
each request



puppetd  
client

Configuration is  
generated from  
*Facts* and the  
*Manifest*  
each request





# Why we chose Puppet:

- ✦ Single base image
- ✦ Central Git Repo
- ✦ 'Extends' MCX
- ✦ Your state - guaranteed
- ✦ Self-documenting
- ✦ Idempotent





# Sample Fact

```
Factor.add("computername") do  
  confine :kernel => :darwin  
  setcode do  
    %x{scutil --get ComputerName}.chomp  
  end  
end
```



# Puppet Syntax

```
file { "/etc/motd":  
  content => "Message of the Day",  
}
```

```
file { "/etc/sudoers":  
  owner  => "root",  
  group  => "wheel",  
  mode   => 0440,  
}
```



# Puppet Syntax

```
$facter      = "facter-1.5.8.dmg"
$pkg_base    = "http://puppet.server.com/pkgs"

package {"$facter":
  source      => "$pkg_base/$facter",
  before      => Package["$puppetcurrent"],
}
```



# puppet resource (i.e. ralsh)

- ✦ Formerly known as ralsh, or “Ruby Abstraction Layer Shell”
- ✦ Tool to inspect a running system
- ✦ Outputs Puppet code

```
bash-3.2# puppet resource group admin
group { 'admin':
  ensure => 'present',
  members => ['root', 'admin', 'glarizza'],
  gid => '80'
}
```



# DEMO



# For More Information

- ✦ The Book “Pulling Strings with Puppet”
  - ✦ Expect an update/refresh after January
- ✦ <http://puppetlabs.com>
- ✦ [\*\*http://links.huronstudents.com/puppet\*\*](http://links.huronstudents.com/puppet) - my wiki
- ✦ <http://groups.google.com/group/puppet-users>
- ✦ <http://glarizza.posterous.com>
- ✦ #puppet on freenode



# Contacting Me

- ✦ glarizza on Twitter
- ✦ [glarizza@me.com](mailto:glarizza@me.com)
- ✦ Walk into Ohio and ask for Gary (it's a small place)

