Hierarchical Systems Policy Management in a Puppet/LDAP Environment

Patrick Paul
Puppet Camp LA – February 22, 2013
Format

- The Then:
  Design inspirations from hierarchical management (AD)
- The Now:
  Taking those same methods while using Puppet
- The How:
  How this was implemented

- Q&A, Nuts and Bolts, Where it could go next
- Code at https://github.com/jaxon6/inheritable-puppet-ldap
What Came Before

- Active Directory, WinAthena (win.mit.edu)
- Computer objects are managed in an LDAP tree
- Policies are applied to the computer objects
Introducing Puppet

- Policies Defined with Puppet Classes

```
class tomcat6::tomcat6 {
  if defined(Class['java::java']) !- true {
    include java::java
  } else {
    #notify { "java::java for tomcat is already defined"; }
  }
  case @labdistid {
    'Ubuntu': {

```

- More Defined Types = More Classes
Associate Classes With Servers

- **Site.pp/Nodes.pp**
  ```ruby
  node 'ldapmaster.domain.com' inherits basenode {
    include s_ldap::master
  }
  node 'humanresources.domain.com' inherits basenode {
    include c_humanresources
  }
  ```

- **External Node Classifier**
  - Puppet Master calls executable, executable looks up node, returns YAML
  - Node data stored in database or other object store

- **LDAP**
Puppet And LDAP

- Apache LDAP Browser makes it easy

- Manually applying classes to servers is painful
  - As is implementing logic to automatically apply classes

- Using LDAP, you can make it intrinsic and inherited
  - It is intrinsic, because classes are based on where the object lives
  - It is inherited, because all child objects inherit the classes of the parent objects
LDAP Tree, With Puppet

- puppetHierarchify.pl and cn=puppetClasses

info attribute, how puppetHierarchify.pl tracks changes
Easy To Use

- Graphical tool for easy in/easy out work
- If you move an object, classes automatically add/remove themselves
- To create a new role, create a new OU and specify `cn=puppetClasses` value
- Add your servers and the classes apply automatically
- Different OUs with different permissions
From First Boot To Fully Installed

- **For bare-metal:**
  - Add DHCP/DNS entries for host(s)
  - PXE-boot server, install OS and auto-enable Puppet
  - Puppet takes it from there

- **For EC2 (using un-modified vendor AMIs):**
  - UserData bash script - sets hostname then runurl script
    - script installs puppet, sets domain-name
  - Puppet takes it from there

- Shell script queries LDAP, appending to autosign.conf
puppetHierarchify.pl And Next Steps

- Make puppetVars inherited
- puppetVar nagioscheck (e.g., nagioscheck=tomcat6)
- All nagioscheck puppetVars automatically apply to server
- Script on Nagios server queries LDAP and automatically applies those checks as defined by nagioscheck
- Other AD features, like inheritance blocking
Thanks/Q&A

- Mediaplatform.com
  - Puppet DevOps person
  - QA Lead

- Puppet Labs

- SCaLE

- Questions and Answers