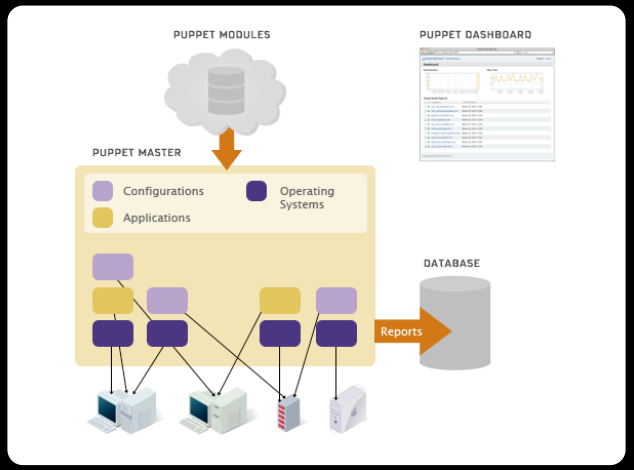
Using Puppet to Create Your Robot Army

Gerard Hickey
January 29, 2011
Macworld 2011 Conference
IT871

What is Puppet?

Puppet is an open source project for automation and configuration management with a declarative language and a flexible client-server architecture

Puppet Architecture



Who Uses Puppet?

Puppet Features

- Cross platform tool
- Centralized server(s) for configurations
- Web based reporting console
- Public key infrastructure for secure communications
- Community module library

Requirements

- Linux, Solaris, BSD or OS X
- Ruby 1.8.4

Installation

- Download distribution files from http://www.puppetlabs.com/misc/download-options
  - Puppet 2.6.4
  - Facter 1.5.8
  - MCollective 1.0.0 (if desired)

Installation

- Installation is as simple as extracting the source file and running the ruby install script (facter does need to be installed first)

```
% tar xzf puppet-2.6.4.tar.gz
% cd puppet-2.6.4
% sudo ruby install.rb
```

Creating the puppetmaster

- Puppet configuration files are in `/etc/puppet`

- Generate basic configuration with

```
puppet master --genconfig > /etc/puppet/puppet.conf
```

Creating the puppetmaster

- Naming your puppetmaster "puppet" is not neccessary, but use FQDN!
  - Will need to modify reportserver, ca_server, server accordingly
  - Default for report_server is server (probably have to comment)

Creating the puppetmaster

- puppetmasterd is not able to make the puppet user and group on OS X
- Create a locked account called puppet and a group also called puppet
- Create OD MCX to deny puppet from logging in and showing up on login panel

Creating the puppetmaster

- Create `/Library/LaunchDaemons/com.puppetlabs.puppetmasterd.plist`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Disabled</key>
        <false/>
        <key>Label</key>
        <string>com.puppetlabs.puppetmasterd</string>
        <key>Program</key>
        <string>/usr/sbin/puppetmasterd</string>
        <key>RunAtLoad</key>
        <true/>
        <key>ServiceDescription</key>
        <string>Launchd script for starting puppetmasterd</string>
        <key>StartInterval</key>
        <integer>5</integer>
</dict>
</plist>
```

- `launchctl load <path to plist>`

Adding a puppet Client

- For the first client execute:
  `sudo puppetd --server <SERVER> --waitforcert 60 --test`
- On puppetmaster, see cert to be signed
  `sudo puppetca --list`
- Sign cert to have client load initial manifest
  `sudo puppetca --sign <CLIENT>`

Adding a puppet Client

- Normally a client just executes puppetd
- The default time for waiting for the cert is 2 minutes
- puppetd will keep checking server for signed cert and then load the manifests

Adding a puppet Client

- Create `/Library/LaunchDaemons/com.puppetlabs.puppetd.plist`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Disabled</key>
        <false/>
        <key>Label</key>
        <string>com.puppetlabs.puppetd</string>
        <key>Program</key>
        <string>/usr/sbin/puppetd</string>
        <key>RunAtLoad</key>
        <true/>
        <key>ServiceDescription</key>
        <string>Launchd script for starting puppetd</string>
</dict>
</plist>
```

- `launchctl load <path to plist>`

Adding a puppet Client

- Generate puppet.conf

```
puppet agent --genconfig > /etc/puppet/puppet.conf
```

- Set server value to puppetmaster
- Paths may be incorrect. May be easier:

```
[agent]
     server = <PUPPETMASTER>
```

- Again use FQDN!

Puppet Language

- Define a series of resources
- Classes function as conditional statements
- Resources can be grouped together

Reference Sites

- Pulling Strings with Puppet
  - http://tinyurl.com/puppet-mw

- http://www.puppetlabs.com/