Satellite 6 and Puppet

Michael Heldebrant
Senior Cloud Infrastructure Solutions Architect
Red Hat
April 2015

AGENDA

- How does Satellite 6 manage puppet

- Steps to get modules available to hosts

- Parameters and Inheritance

- Smart Matchers

- Validate

Trivia Question:

Bunraku - Uemura Bunrakuken came to Osaka from Awaji and began his own theater

Originally, the term Bunraku referred only to the particular theater established in 1805.

Also known as: Ningyō jōruri (人形浄瑠璃)

Satellite 6 Components

https://puppetlabs.com/puppet/puppet-open-source

Open Source Puppet is a declarative, model-based configuration management solution that lets you define the state of your IT infrastructure, using the Puppet DSL.

Open Source Puppet then automatically enforces the correct configuration, making sure the right services are up and running.

By automating these manual tasks, you free up time to work on projects that deliver greater business value.

Satellite 6: Hosts->All hosts

A Host is a Foreman concept that represents a server/host/system/computer. In addition to holding facts about the system, it:

- Stores which operating system the system should be running
- Stores which puppet classes should be assigned
- Stores which parameters apply to which puppet classes
- Allows you to re-provision the machine
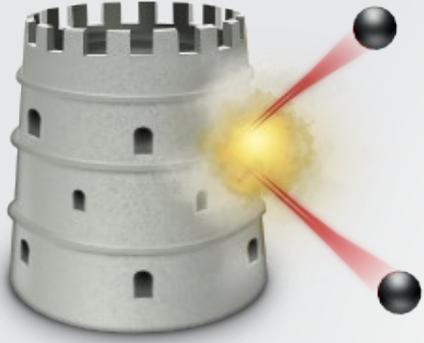
What is Katello?

Katello brings the full power of content management alongside the provisioning and configuration capabilities of Foreman.
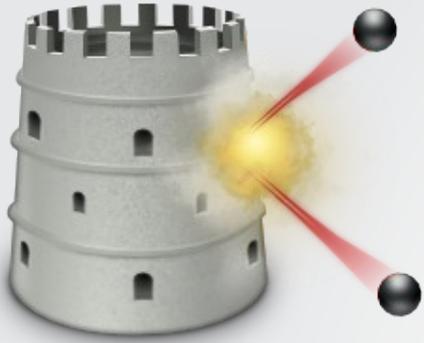
Satellite 6: Hosts->Content Hosts

Content Hosts are the part of a host that manages Content and Subscription related tasks.
As time goes on more and more of this functionality will be moved to the Host object. A Host's Content Host:

    Stores which Products are assigned (i.e. which Repositories will the system pull content from)
    Initiates package install/upgrade/removal
    Determines which errata apply to a system
    Initiates errata installation

How is puppet managed in Satellite 6

Puppet module content is stored in pulp and managed by the katello portion of Satellite 6

Satellite 6 provides external data to the puppet master via the enc interface

Puppet environments are mapped directly in the foreman portion of Satellite 6

Puppet environments are generally used to separate classes from different types of Hosts which allows you to use different modules in different environments

redhat.
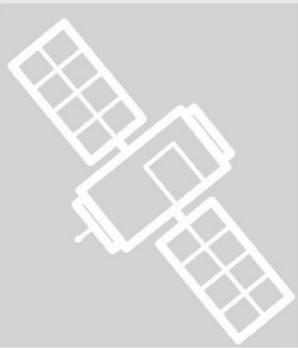
Step by Step from module to managed host

Server:

Get puppet modules into Satellite 6

Provide puppet modules in a content view

Select modules from that content view via a host group, host, etc to apply to hosts

Configure default values and override values by criteria

Clients:

Install puppet (from rhel-7-server-rh-common-rpms)

Configure /etc/puppet/puppet.conf to point at the Satellite

Part of host provisioning in Satellite 6 via provisioning template: Satellite Kickstart Default

Store puppet modules in Content->products

Get puppet modules from Puppet Forge

Build puppet modules

Importing your Puppet Modules from a Git repo

Use puppet module generate to build the correct metadata files and directory structure for modules

Run pulp-puppet-module-builder (rpm: pulp-puppet-tools)

checkout of the repository and branch
builds all of the modules
publishes them in a structure Katello can synchronize either local or via http server

Select and Provide modules in Content Views

Composite Content Views

Select puppet environment in host groups

Edit RHEL 7 SALAB Puppet Guests

| Host Group | Puppet Classes | Network | Operating System | Parameters | Locations | Organizations | Activation Keys |

Parent

Name RHEL 7 SALAB Puppet Guests

Lifecycle Environment Dev

Puppet Environment Docker_host_view

redhat.

Select modules in host groups (or hosts, etc)

Puppet Environment has classes from content view

Puppet classes

environment = KT_Default_Organization_Dev_Docker_host_view_48   **Q Search** ▾   Import from satellite.rhsat.rdu.salab.redhat.com   **New Puppet class**

Class name	Environments and documentation	Host group	Hosts	Parameters	Variables	
motd	KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48 KT_Default_Organization_Library_DEMO_Puppet_54 KT_Default_Organization_RHEL_WS_Test_RHEL_6_colin_29 KT_Default_Organization_Prod_Retail_Tomcat1_RHEL6_TOMCAT_RRIOS_51 KT_Default_Organization_QA_Retail_Tomcat1_RHEL6_TOMCAT_RRIOS_51 KT_Default_Organization_Dev_Retail_Tomcat1_RHEL6_TOMCAT_RRIOS_51 KT_Default_Organization_Library_RHEL6_TOMCAT_RRIOS_51 KT_Default_Organization_RHEL_WS_Dev_RHEL_6_colin_29 KT_Default_Organization_Library_RHEL_6_colin_29 KT_Default_Organization_Dev_RHEL7_amd_32 KT_Default_Organization_Library_RHEL7_amd_32 KT_Default_Organization_Retired_matt_test_15 KT_Default_Organization_Demo1_matt_test_15 KT_Default_Organization_Library_matt_test_15	RHEL 7 SALAB Puppet Guests and RHEL WS	0	20	0	Delete
ntp	KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48 KT_Default_Organization_Library_DEMO_Puppet_54	RHEL 7 SALAB Puppet Guests	0	24	0	Delete
ntp::config	KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48 KT_Default_Organization_Library_DEMO_Puppet_54		0	0	0	Delete
ntp::install	KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48 KT_Default_Organization_Library_DEMO_Puppet_54		0	0	0	Delete
ntp::params	KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48 KT_Default_Organization_Library_DEMO_Puppet_54		0	0	0	Delete
ntp::service	KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48 KT_Default_Organization_Library_DEMO_Puppet_54		0	0	0	Delete

KT_Default_Organization_Library_Docker_host_view_48 KT_Default_Organization_Dev_Docker_host_view_48

Parameters – Inheritance part 1

Global parameters inheritance

Globally defined parameters
    Configure > Global parameters

Organization-level parameters
    Administer > Organizations > edit > Parameters

Location-level parameters
    Administer > Locations > edit > Parameters

Parameters – Inheritance part 2

Domain-level parameters
    Infrastructure > Domains > edit > Parameters
Operating system-level parameters
    Hosts > Operating systems > edit > Parameters
Host group-level parameters
    Configure > Host groups > edit > Parameters
Host parameters
    Hosts > All hosts > edit > Parameters

Edit a Host and switch to the Parameters, and you will see all of its inherited parameters from the previous levels. Note that they will all be marked as "Scope: Global" as this refers to the Puppet scope, not the Foreman scope. You can override any of these previously-defined parameters or define new ones here.

Smart Variables and Smart Parameters

Smart variables are a tool to provide global parameters (key/value data), normally to your Puppet ENC, depending on a set of rules.

They are intended to be a stepping stone to full parameterized classes, when the class hasn't been parameterized or in special cases when a global parameter is desired

Smart variables are associated with a Puppet class, but they result in a global parameter. They may have multiple possible values, all depending on hierarchical context or various conditions a user can wish to apply.

Smart parameters allow a puppet class to request external data

If a class needs to configure itself with data other than facts, that data should usually enter the class via a parameter.

Parameters are essentially the API to use puppet modules.

Smart Matchers Overview

Smart matching technology manages both smart variables and smart class parameters:

A default value that can be sent if no specific match is found.

An order of precendence for overrides, based on host attributes or facts.

A list of overrides (matchers).

Specifying a data type, allowing strings, integers and data structures to be passed natively to Puppet.

Optional validation of values.

Template processing of values for dynamic content.

Smart Matchers Default Value

Most importantly, the Override option has to be enabled for Foreman to control this variable, otherwise it will never be managed and will not appear in the ENC output.

The Default value will be supplied in the ENC output and should be a supported value, such as a string, YAML or JSON structure or use template features (see following sections). When the Use Puppet default checkbox is enabled, no default value will be present in the ENC output unless an override matches.

Smart Matchers Ordering

Overrides are processed in the order of precedence from most to least specific

Example attributes:
fqdn   - host's FQDN ("host.example.com")
hostgroup   - full name including parents ("Europe/Web servers")
os   - name and version ("RedHat 6.4")
domain   - host's domain name ("example.com")
location or organization - full name including parents ("Company/Subsidiary")
is_virtual   - a fact supplied by Facter

The default order is "fqdn", "hostgroup", "os", "domain"
        Adminster > Settings > Puppet > Default_variables_Lookup_Path

Smart Matchers Overrides

Add criteria to match against - click the Add Matcher-Value button under your parameter, and more input fields will appear:

Match   Should state a name = value relationship to match against the entries in the order list
Value   What the parameter should be in the ENC, if this rule is matched

Smart Matchers Templates

Dynamic Data is possible by using foreman parameters and puppet facts

See:
http://projects.theforeman.org/projects/foreman/wiki/templatewriting

Match	hostgroup=RHEL 7 SALAB Puppet Guests

ⓘ Explain matchers

Value	==== Test for Docker host <%= @host.facts_hash['hostname'] %> OS <%= @host.facts_hash['operatingsystem'] %> <%= @host.facts_hash['operatingsystemrelease'] %> ==== ✖

redhat.

Smart Matchers Can Execute Code

Possible to use ruby code in the template expansions:

Conditionals
<% if @host.hostgroup.to_s == "Base/Application Servers" ->...< end -%>

Loops
<% @host.interfaces.each do |i| %> key is <%= i.ip %> <% end %>

Puppet ERB tags

https://docs.puppetlabs.com/guides/templating.html

    <%= Ruby expression %> — This tag will be replaced with the value of the expression it contains.

    <% Ruby code %> — This tag will execute the code it contains, but will not be replaced by a value. Useful for conditional or looping logic, setting variables, and manipulating data before printing it.

    <%# comment %> — Anything in this tag will be suppressed in the final output.

    <%% or %%> — A literal <% or %>, respectively.

    <%- — Same as <%, but suppresses any leading whitespace in the final output. Useful when indenting blocks of code for readability.

    -%> — Same as %>, but suppresses the subsequent line break in the final output. Useful with many lines of non-printing code in a row, which would otherwise appear as a long stretch of blank lines.

Validate the data for a host

To see how Foreman is passing the parameters to Puppet, go to a Host and click the YAML button: the exact YAML data sent to the Puppet master

```
---
classes:
  motd:
    motd_content: ! "====\r\nTest for Docker host mheldebr-docker3 OS RedHat 7.0\r\n===="
  ntp:
parameters:
  puppetmaster: satellite.rhsat.rdu.salab.redhat.com
  domainname: ''
  hostgroup: RHEL 7 SALAB Puppet Guests
  location: Default_Location
  organization: Default_Organization
  root_pw: REDACTEDHASH
  puppet_ca: satellite.rhsat.rdu.salab.redhat.com
  foreman_env: KT_Default_Organization_Dev_Docker_host_view_48
  owner_name: Admin User
  owner_email: root@rhsat.rdu.salab.redhat.com
  kt_activation_keys: AK-Reg_to_Dev
  kt_cv: Docker_host_view
  kt_env: Dev
  kt_org: Default_Organization
environment: KT_Default_Organization_Dev_Docker_host_view_48
```