



Enterprise-Grade Open-Source Network Management

Who Pulls the Strings?

Integrating OpenNMS with Modern Configuration Management

FOSDEM 2012, 05 Feb 2012

Jeff Gehlbach

jeffg@opennms.org

Ronny Trommer

ronny@opennms.org

Network Management

Routers



Switches



Firewalls

Think fast.®



HUAWEI

Load balancers

WAN accelerators



VPN concentrators

... does it have an IP address?*

Systems Management

Major distributions of Linux 2.6 and later

Mac OS X



*BSD

(Open)?(Solaris|Indiana) 10+

Similar but different: AIX, HP-UX

Even Windows!



Monitoring and Managing

Big problem domain when many nodes
are involved!

How to add them all to be managed?

Services up and responding quickly?

Something happened, how to know?

What's happening under the hood?

Something shiny to show the boss?

Enter OpenNMS[®]

“World's First Enterprise-Grade Network Management Platform Developed Under the Open Source Model”

Started in 1999 by ex-OpenView hackers

Maintained by the Order of the Green Polo

Supported, sponsored by my employer

Consistent model designed for huge scale

100% GPLv3 codebase

Will never suck

Will always be Free (as in Freedom)

~~Fauxpen
Source~~

Not “Based On *Tool X*”

Built from the ground up

100% GPLv3 code base, Java



Makes extensive use of good libraries

Does not duct-tape in other apps

→ That way lies the end of scalability

→ Not to mention maintainability

Architectural decisions dictated by
requirement to scale **huge**.

Designed to Save You Time

If you want a monitoring app that works just the way you want right out of the box, keep looking. OpenNMS is a platform, not a fixed-function application.

It is designed to “front-load” the effort involved in a given task. The payoff comes in easy, automatic repetition of that task at scale.

Sound familiar?

Use What Works For You

If you're happy, don't mess with it.

But maybe it wasn't designed for that...



Photo credit: Wikimedia Commons
Analogy: Alex Finger <af@genevainformation.ch>

Discovery and Provisioning

Discovery: Awareness of a previously unknown IP address, usually via ping



Image: Wikimedia Commons



Image: Wikimedia Commons

Provisioning: Finding out all we can and representing results in our model.

Service(s) → Interface(s) → Node

Provisioning

Capsd: Legacy capabilities scanner.

Automatic Provisioning: Seed an IP address; scan for interfaces and services.

***Directed Provisioning*: Seed an exact set of known IP interfaces and services.**

Policy-Based Provisioning: Seed an IP address; scan for interfaces and services, deciding on persistence, data collection, service monitoring, categorization...

Provisioning (cont'd)

External provisioning sources...

DNS import: Create nodes and interfaces from A / AAAA records in a zone

ReST API: Push-wise from outside

Your DB: Make a CGI that generates XML describing your systems, feed URL to Provisiond, watch magic happen

This is shouting for a Puppet integration!

Directed Provisioning

- Every node created this way is part of a *requisition* and has:
 - **Foreign Source**: a string that groups a set of nodes; identical to the name of the containing *requisition*. Slightly analogous to Puppet's *environments*.
 - **Foreign ID**: a string that uniquely identifies a node within a requisition.
- *Foreign-Source:Foreign-ID* makes an identifying tuple for a node.

CUE RONNY!

Thoughts for the FOSDEM hack

- How we can get data from puppet?
- What has to be written in OpenNMS?
- Restrictions?
- Further improvements?

```
curl -k -H "Accept: yaml" \  
https://puppetmaster:8140/production/facts_search/search
```

- patches.mydomain.net
- swlab.mydomain.net
- itchy.mydomain.net
- scratchy.mydomain.net
- lvps.mydomain.net

```
1 --- object:Puppet::Node
2 classes: []
3 environment: &id001 production
4 expiration: 2012-02-02 20:45:21.886019 +01:00
5 name: patches.
6 parameters:
7   swapfree: 479.81 MB
8   kernel: Linux
9   netmask: 255.255.255.224
10  physicalprocessorcount: "1"
11  processorcount: "2"
12  lsbmajdistrelease: "10"
13  operatingssystemrelease: "10.04"
14  uniqueid: aec1051d
15  fqdn: patches.
16  clientversion: &id002 2.7.1
17  ipaddress:
18  is_virtual: "false"
19  virtual: physical
20  memorysize: 2.46 GB
21  ps: ps -ef
22  hardwaremodel: i686
23  rubysitedir: /usr/local/lib/
24
```

```
curl -k -H "Accept: yaml" \
https://{puppetmaster}:8140/{environment}/node/{puppetNode}
```

/opt/opennms/etc/imports/production.xml

```
puppet - vim - 111x36
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <model-import last-import="2012-02-02T21:09:46.032+01:00" foreign-source="production" date-stamp="2012-02-
  02T21:09:00.053+01:00" xmlns="http://xmlns.opennms.org/xsd/config/model-import">
3   <node node-label="██████████.dedicated.hosteurope.de" foreign-id="a953332a">
4     <interface status="1" snmp-primary="P" managed="true" ip-addr="██████████"/>
5     <asset value="Intel(R) Xeon(R) CPU E5520 @ 2.27GHz" name="cpu"/>
6     <asset value="" name="modelName"/>
7     <asset value="" name="serialNumber"/>
8     <asset value="Ubuntu 10.04.3 LTS" name="operatingSystem"/>
9     <asset value="" name="manufacturer"/>
10  </node>
11  <node node-label="scratchy.██████████" foreign-id="007f0101">
12    <interface status="1" snmp-primary="P" managed="true" ip-addr="██████████"/>
13    <asset value="Intel(R) Xeon(TM) CPU 3.00GHz" name="cpu"/>
14    <asset value="ProLiant DL360 G4" name="modelName"/>
15    <asset value="GBJ██████████ E" name="serialNumber"/>
16    <asset value="Ubuntu 11.10" name="operatingSystem"/>
17    <asset value="HP" name="manufacturer"/>
18  </node>
```

OPENMS_HOME/etc/provisond-configuration.xml

```
<requisition-def import-name="production"  
    import-url-resource="puppet://puppetmaster:8140/production">  
    <cron-schedule>0 9 21 * * ? *</cron-schedule>  
</requisition-def>
```

opennms-config/src/main/castor/provisiond-configuration.xsd

```
52 <simpleType name="supported-protocols" >
53   <annotation>
54     <documentation>
55       This type specifies the URLs currently supported by the Provisioner with
56       a mild syntax restriction. Currently support URLs: for the protocols: http,
57       https, file, dns
58     </documentation>
59   </annotation>
60   <restriction base="string" >
61     <pattern value="(puppet dns|file|http|https)://.*">
62   </pattern>
63   </restriction>
64 </simpleType>
65
66 </schema>
```

~/Develop/workbench-puppet/IDE-puppet/opennms-config/src/main/castor/provisiond-configuration

opennms-util/src/main/java/org/opennms/core/utils/url/ GenericURLFactory.java

```
19 private GenericURLFactory() {
20     urlConnections = new HashMap<String, String>();
21
22     // Map dns:// against DNS requisition URL connection
23     urlConnections.put("dns", "org.opennms.netmgt.provision.service.dns.DnsRequisitionURLConnection");
24     logger.debug("Add dns protocol to map against org.opennms.netmgt.provision.service.dns.DnsRequisitionURLConnection");
25
26     // Map puppet:// against Puppet requisition URL connection
27     urlConnections.put("puppet", "org.opennms.netmgt.provision.service.puppet.PuppetRequisitionURLConnection");
28     logger.debug("Add puppet protocol to map against org.opennms.netmgt.provision.service.puppet.PuppetRequisitionURLConnection");
29 }
30
31 public static void initialize() {
32     if (genericUrlFactory == null) {
33         genericUrlFactory = new GenericURLFactory();
34         URL.setURLStreamHandlerFactory(genericUrlFactory);

```

opennms-provision/opennms-provisiond/
src/main/java/org/opennms/netmgt/provision/service

```
2 .
3 |— TimeTrackingMonitor.java
4 |— WorkDuration.java
5 |— WorkEffort.java
6 |— dns
7 |   |— DnsRequisitionURLConnection.java
8 .
9 .
10 |— puppet
11 |   |— PuppetRequisitionURLConnection.java
12 |   |— PuppetRestClient.java
13 |   |— SSLUtilities.java
14 |— snmp
15 |   |— IfTable.java
16 |   |— IfTableEntry.java
17 .
18 .
```

Maven Dependencies

- I used Jersey API for ReST – its already in the project
- Add snakeYAML dependency in opennms-provisiond/pom.xml

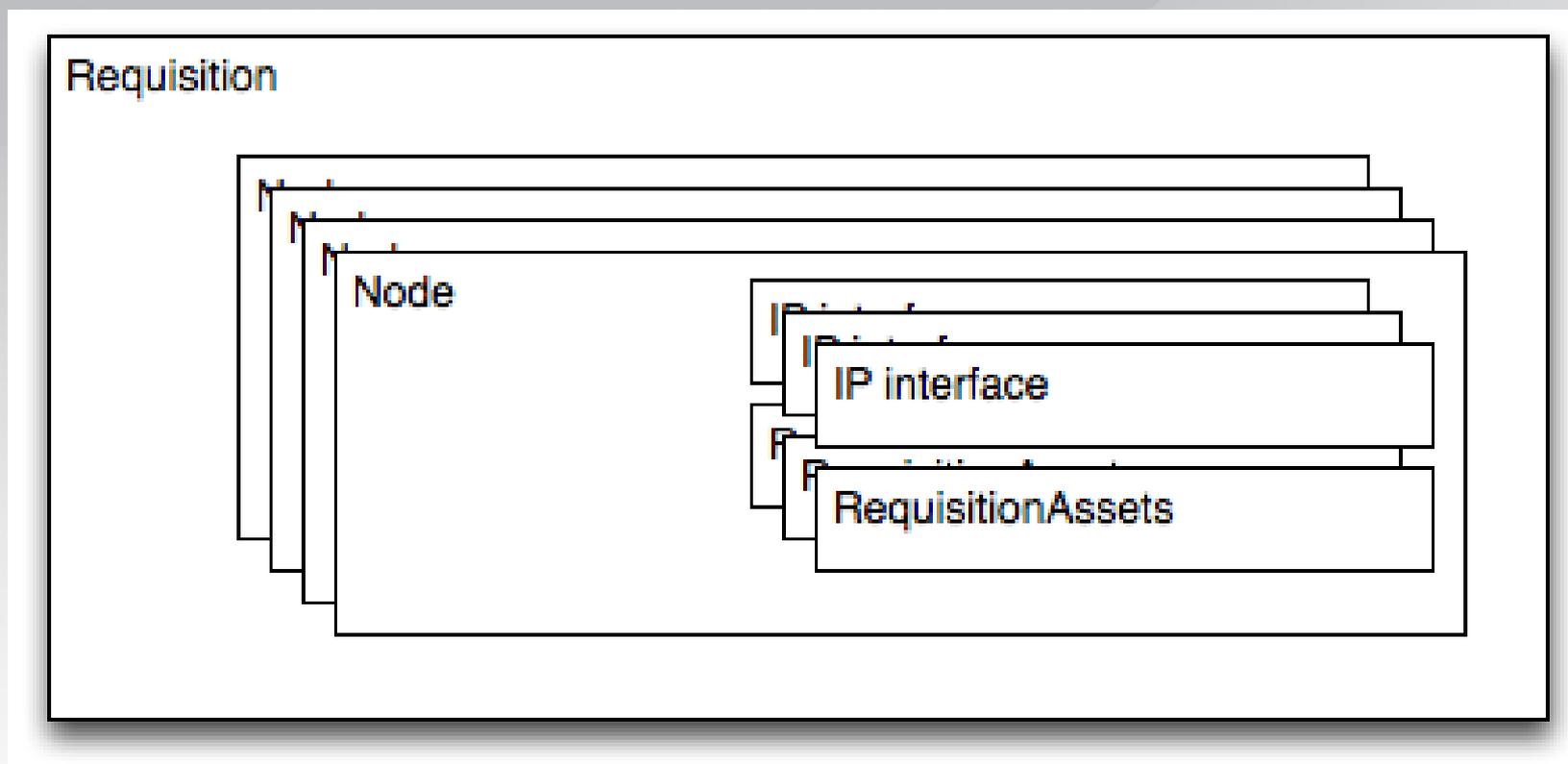
```
<dependency>  
  <groupId>org.yaml</groupId>  
  <artifactId>snakeyaml</artifactId>  
  <version>1.9</version>  
</dependency>
```

PuppetRequisitionURLConnection.java

- create a class and extend `GenericURLConnection`
- We override `getInputStream()` – Does all the important stuff
- You have to implement `connect()` – We don't need it, make a NOP :)
- Create a Class for the `PuppetRestClient` and some helper methods

PuppetRequisitionURLConnection.java

You have to build this structure



Things we have to do

- Create a new Requisition with foreign-source
- Request the nodes you want to import from Puppet
- Iterate over each node and get the facts from puppet
- Create and fill up for each node a RequisitionNode object
- Insert the interface to the RequisitionNode
- Assign the interface to a node and set it to Primary, Secondary or Non for SNMP data collection
- Fill up and assign RequisitionAssets to the node
- Insert the filled RequisitionNode into the Requisition
- Return it as XML stream for the Provisioner

Some rules

- #1 Set a name for the foreign-source
- #2 Set a node label for each node
- #3 Set at minimum one IP interface
- #4 Set *one* IP-Interface as primary interface for SNMP data collection
- #5 If you have more than one IP-Interface, you have to set them to “secondary” or “non”

PuppetRequisitionUrlConnection

```
private Requisition buildPuppetRequisition() {
    Requisition requisition = new Requisition(m_foreignSource);

    try {
        // m_puppetRestUrl should be https://{puppetmaster}:8140
        m_puppetRestClient = new PuppetRestClient(m_puppetRestUrl);
        List<String> puppetNodeList = m_puppetRestClient.getPuppetNodesByFactsSearch(m_puppetEnvironment, m_factSearch);

        for (String puppetNode : puppetNodeList) {
            requisition.insertNode(createRequisitionNode(puppetNode));
            logger.debug("Insert puppet requisition node '{}' from environment '{}'", puppetNode, m_puppetEnvironment);
        }
    } catch (URISyntaxException e) {
        e.printStackTrace(); //To change body of catch statement use File | Settings | File Templates.
    }

    return requisition;
}
```

Create and fill up the node

```

private RequisitionNode createRequisitionNode(String puppetNode) {
    logger.debug("Create requisition node for puppet node '{}'", puppetNode);
    // Initialize a node
    RequisitionNode requisitionNode = new RequisitionNode();

    // Initialize a interface
    RequisitionInterface requisitionInterface = new RequisitionInterface();

    // environment -> Puppet environment: production
    // puppetNode -> Puppet node name: itchy.opennms-edu.net
    Map<String, String> puppetNodeFacts = m_puppetRestClient.getFactsByPuppetNode(m_puppetEnvironment, puppetNode);

    // Set OpenNMS node label
    requisitionNode.setNodeLabel(puppetNode);
    logger.debug("Set node label: '{}'", puppetNode);

    // Foreign Id as unique ide in puppet master
    // TODO indigo: We have to be sure "uniqueid" is really unique in puppet
    requisitionNode.setForeignId(puppetNodeFacts.get("uniqueid"));
    logger.debug("Set node foreign ID: '{}'", puppetNodeFacts.get("uniqueid"));

    // Verify IP addresses an initialize the interface model
    // TODO indigo: We have to figure out what happen with more then one ip interface and IPv6?
    try {
        InetAddress inetAddress = InetAddress.getByAddress(puppetNodeFacts.get("ipaddress"));
        requisitionInterface.setIpAddr(puppetNodeFacts.get("ipaddress"));
        requisitionInterface.setSnmpprimary("P");
    } catch (UnknownHostException e) {
        logger.error("Error parsing IP address '{}'. Error message: '{}'", puppetNodeFacts.get("ipaddress"), e.getMessage());
    }

    // Assign puppet facts to OpenNMS assets
    RequisitionAsset manufacturerAsset = new RequisitionAsset("manufacturer", puppetNodeFacts.containsKey("manufacturer") ? puppetNodeFacts.get("manufacturer") :
    RequisitionAsset osAsset = new RequisitionAsset("operatingSystem", puppetNodeFacts.containsKey("lsbdistdescription") ? puppetNodeFacts.get("lsbdistdescription") :
    RequisitionAsset serialNumberAsset = new RequisitionAsset("serialNumber", puppetNodeFacts.containsKey("serialnumber") ? puppetNodeFacts.get("serialnumber") :
    RequisitionAsset modelNameAsset = new RequisitionAsset("modelName", puppetNodeFacts.containsKey("productname") ? puppetNodeFacts.get("productname") : "");
    RequisitionAsset cpuAsset = new RequisitionAsset("cpu", puppetNodeFacts.containsKey("processor0") ? puppetNodeFacts.get("processor0") : "");

    // Add assets to requisition node
    requisitionNode.putAsset(manufacturerAsset);
    requisitionNode.putAsset(osAsset);
    requisitionNode.putAsset(serialNumberAsset);
    requisitionNode.putAsset(modelNameAsset);
    requisitionNode.putAsset(cpuAsset);

    // Configure the interface and initialize
    requisitionInterface.setManaged(Boolean.TRUE);
    requisitionInterface.setStatus(Integer.valueOf(1));
    requisitionNode.putInterface(requisitionInterface);

    return requisitionNode;
}

```

PuppetRestClient

```
public PuppetRestClient(URL url) throws URISyntaxException {  
  
    // If you want base authentication go for defaultApacheClientConfig and credentials  
    m_clientConfig = new DefaultClientConfig();  
    m_client = Client.create(m_clientConfig);  
  
    logger.debug("Initialize Puppet ReST client with URL: '{}'", url.toString());  
    m_webResource = m_client.resource(url.toURI());  
}
```

PuppetRestClient

```
public List<String> getPuppetNodesByFactsSearch(String environment, String search) {
    Yaml puppetNodeListYaml = new Yaml();
    ArrayList<String> puppetHosts = new ArrayList<>();

    // https://{puppetmaster}:8140/{environment}/facts_search/search?{search}
    // environment = production
    // search = facts.operatingsystem=Ubuntu
    // production/facts_search/search?facts.productname=VMware%20Virtual%20Platform
    String puppetSearchResult = m_webResource.path(environment).path("facts_search")
        .path("search?" + search).accept(MEDIA_TYPE_YAML).get(String.class);
    logger.debug("Search result with search '{}' for puppet nodes: '{}'", "search?" +
        search, puppetSearchResult);

    puppetHosts = (ArrayList<String>) puppetNodeListYaml.load(puppetSearchResult);
    return puppetHosts;
}
```

PuppetRestClient

```
public Map<String,String> getFactsByPuppetNode(String environment, String puppetNode) {
    Map<String,String> nodeFacts = new HashMap<>();

    //https://{puppetmaster}:8140/{environment}/node/{puppetNode}
    String puppetNodeYaml = m_webResource.path(environment).path("node")
        .path(puppetNode).accept(MEDIA_TYPE_YAML).get(String.class).replace("!ruby/", "");
    logger.debug("Get puppet node facts for node '{}': '{}'", puppetNode, puppetNodeYaml);

    try {
        ArrayList<String> lines = (ArrayList<String>) IOUtils.readlines(new StringReader(puppetNodeYaml));
        for (String line : lines) {
            nodeFacts.put(line.split(":")[0].trim(), line.split(":")[1].trim().replaceAll("\\\"", ""));
        }
    } catch (IOException e) {
        e.printStackTrace(); //To change body of catch statement use File | Settings | File Templates.
    }
    return nodeFacts;
}
```

PuppetRestClient

```
public Map<String,String> getFactsByPuppetNode(String environment, String puppetNode) {
    Map<String,String> nodeFacts = new HashMap<>();

    //https://{puppetmaster}:8140/{environment}/node/{puppetNode}
    String puppetNodeYaml = m_webResource.path(environment).path("node")
        .path(puppetNode).accept(MEDIA_TYPE_YAML).get(String.class).replace("!ruby/", "");
    logger.debug("Get puppet node facts for node '{}': '{}'", puppetNode, puppetNodeYaml);

    try {
        ArrayList<String> lines = (ArrayList<String>) IOUtils.readlines(new StringReader(puppetNodeYaml));
        for (String line : lines) {
            nodeFacts.put(line.split(":")[0].trim(), line.split(":")[1].trim().replaceAll("\\\"", ""));
        }
    } catch (IOException e) {
        e.printStackTrace(); //To change body of catch statement use File | Settings | File Templates.
    }
    return nodeFacts;
}
```

This part SUCKS!

Create and return the XML output

```
@Override
public InputStream getInputStream() throws IOException {

    InputStream stream = null;

    try {
        Requisition r = buildPuppetRequisition();
        stream = new ByteArrayInputStream(JaxbUtils.marshal(r).getBytes());
    } catch (Throwable e) {
        logger.error("Problem getting input stream: '{}'", e);
        throw new IOExceptionWithCause("Problem getting input stream: " + e, e);
    }

    return stream;
}
```

Home / Admin / Provisioning Requisitions / Edit Requisition

Requisitioned Nodes for Group: production

Done Add Node

- Node **lvns** ForeignId **a953332a** Site [Add Interface] [Add Node Category] [Add Node Asset]
 - IP Interface Description SNMP Primary **P** Add Service
 - asset **cpu** Intel(R) Xeon(R) CPU
 - asset **modelName**
 - asset **serialNumber**
 - asset **operatingSystem** Ubuntu 10.04.3 LTS

- Node **scratchy** ForeignId **007f0101**
 - IP Interface
 - asset **cpu** Intel(R) Xeon(TM) CPU 3.0
 - asset **modelName** ProLiant DL360 G4
 - asset **serialNumber** GB
 - asset **operatingSystem** Ubuntu 11.10
 - asset **manufacturer** HP

Notification details

Home / Notification / Detail

Notice #1 from event #2711

Notification Time	2/2/12 9:16:23 PM	Time Replied		Responder	
Node	patches.opennms- edu.net	Interface		Service	ICMP

See outages for patches.opennms-[patches.opennms-
edu.net](#)

Numeric Message

111-1

Text Message

All services are down on node patches.opennms-[patches.opennms-
edu.net](#). New Outage records have been created and service level availability calculations will be impacted until this outage is resolved. Modelnumber: ProLiant DL380 G3 Serialnumber: 80LYKJN21D

Operatingsystem: Ubuntu 10.04.3 LTS

Sent To	Sent At	Media	Contact Info
admin	2/2/12 9:16:25 PM	javaEmail	

Notification configuration

[Home](#) / [Admin](#) / [Configure Notifications](#) / [Choose Path](#)

Editing notice: nodeDown

Choose the destination path and enter the information to send via the notification

Name:	<input type="text" value="nodeDown"/>
Description:	<input type="text"/>
Parameter:	Name: <input type="text"/> Value: <input type="text"/>
Choose A Path:	<input type="text" value="Email-Admin"/>
Text Message:	<p>All services are down on node %nodelabel%. New Outage records have been created and service level availability calculations will be impacted until this outage is resolved.</p> <p>Modelnumber: %asset[modelnumber]% Serialnumber: %asset[serialnumber]% Operatingsystem: %asset[operatingsystem]%</p>

Restrictions and Improvements

- Currently we don't have a “Java-Puppet-Node-Model”
- WEBrick with ReST and scalability
<http://bitfieldconsulting.com/scaling-puppet-with-distributed-version-control>
- Filter import for Nodes based on a fact search like `search?facts.productname=bla`
- if (possible) ? One ReST call for nodes and facts : leave it at it is

This could be helpful

Puppet

- http://docs.puppetlabs.com/guides/rest_auth_conf.html
- http://docs.puppetlabs.com/guides/rest_api.html

OpenNMS

- http://www.opennms.org/wiki/Developing_with_Git
- http://opennms.org/wiki/Eclipse_and_OpenNMS
- http://www.opennms.org/wiki/IDEA_and_OpenNMS
- <http://www.opennms.org/wiki/Installation:Source>
`git checkout -b feature-puppet origin/feature-puppet`
- irc.freenode.org – #opennms
- http://www.opennms.org/wiki/Mailing_lists

Contacts

- ronny@opennms.org
- IRC-Nick: `_indigo`
- Hit me if you can shed some light to build a nice POJO from Puppets YAML output

Pull-to-Puppet Approach

- Still a work in progress
- External Node Classifier for OpenNMS
 - Iterate the OpenNMS Nodes ReST Service using HTTParty or equivalent gem
 - Output YAML to populate puppetmaster
 - Jason Aras has written a prototype
 - <https://gitorious.org/opennms-puppet-node-pusher>
 - It's called a pusher yet it pulls. I know :p
 - We are not Rubyists, don't ask us detailed questions about the following slides :)

Pull-to-Puppet: Code PLZ?

```
1 require 'rubygems'
2 require 'httparty'
3
4 require 'pp'
5 # l/p/url
6
7 $user = 'admin'
8 $password = 'admin'
9 $base_url = "http://opennms:8980/opennms/rest/"
10
11 class Nodes
12   include HTTParty
13   base_uri $base_url
14   basic_auth $user, $password
15   format :xml
17 end
18
19 response = Nodes.get('/nodes', :query => {:limit => 0})
20
21 x = response.parsed_response
22
23 x["nodes"]["node"].each do |node|
24   if node["label"] == ARGV[0]
27     comments = node['assetRecord']['comment']
29
30     comments.each do |line|
31       line.strip!
32
33       if line.lstrip.match(/^puppet/) && ((line.include? "environment") || (line.include?
"parameters") || (line.include? "classes"))
```

Pull-to-Puppet: MOAR Code!!!

```
33     if line.lstrip.match(/^puppet/) && ((line.include? "environment") || (line.include?
"parameters") || (line.include? "classes"))
34         #puts line
35
36         if (line.include? "environment")
37             @environment = line[line.index(':')+1..line.length].strip
38
39         elsif (line.include? "classes")
40             @classes = line[line.index(':')+1..line.length].split(',')
41             @classes.collect! { |x| x.strip!}
42         elsif (line.include? "parameters")
43             @parameters = {}
44             kvpairs = line[line.index(':')+1..line.length].split(',')
45             kvpairs.each do |str|
46                 (k,v) = str.split("=")
47                 @parameters[k.strip] = v.strip
48
49             end
50         end
51     end
52 end
56 end
57
58 end
59
60 output = {}
61 output['classes'] = @classes
62 output['parameters'] = @parameters
63 output['environment'] = @environment
64 puts output.to_yaml
```

What Just Happened?

- Asked OpenNMS for all its nodes
- Scraped *comments* asset field for each to divine Puppet metadata:
 - Environment name
 - Class names
 - Parameters
- This is quick and dirty code
 - Ideas welcome for clean, configurable strategies mapping data across domains

cartman.internal.opennms.com | Node | OpenNMS Web Console - Mozilla Firefox

File Edit View History Bookmarks Tools Help

cartman.internal.opennms.com | ...

internal.opennms.com:18980/opennms/element/node.jsp?node=9

openNMS®

User: admin (Notices On) - Log out
Feb 4, 2012 18:09 EST

Node List Search Outages Path Outages Dashboard Events Alarms Notifications Assets Reports Charts Surveillance Distributed Status Distributed Map Map Add Node Admin Support

Home / Search / Node
Node: cartman.internal.opennms.com
Not a member of any provisioning requisition
View Events View Alarms View Outages Asset Info HTTP Resource Graphs Rescan Admin Schedule Outage

Asset Information

Description

Comments environment: production classes: debian,opennmsgroup,apacheserver,sambaserver
parameters: ntp_servers=time.ncsu.edu,mail_server=mail1.opennms.com

SNMP Attributes

Name	cartman
Object ID	.1.3.6.1.4.1.8072.3.2.10
Location	Blast Server Room, 220 Chatham Business Drive, Pittsboro, NC, 27312, USA
Contact	Ben Reed
Description	Linux cartman 2.6.26-2-686 #1 SMP Wed Sep 21 04:35:47 UTC 2011 i686

Availability

Availability (last 24 hours)	100.000%
Overall	100.000%
DNS	100.000%
FTP	100.000%
HTTP	100.000%
HTTPS	100.000%
ICMP	100.000%
IMAP	100.000%
SMTP	100.000%
SNMP	100.000%
SSH	100.000%
StrafePing	Not Monitored
Update	100.000%
Overall	100.000%
DNS	100.000%
HTTP	100.000%
HTTPS	100.000%

General (Status: Active)

View Node IP Route Info
View Node Link Detailed Info

Surveillance Category Memberships (Edit)

This node is not a member of any categories.

Notification

You: Outstanding: (Check)
You: Acknowledged: (Check)

Recent Events

<input type="checkbox"/>	181134	2/4/12 16:51:30	Indeterminate	An event with no matching configuration was received from interface 172.20.1.10.
<input type="checkbox"/>	181131	2/4/12 16:51:30	Indeterminate	An event with no matching configuration was received from interface 172.20.1.10.
<input type="checkbox"/>	181034	2/4/12 11:51:29	Indeterminate	An event with no matching configuration was received from interface 172.20.1.10.
<input type="checkbox"/>	181028	2/4/12 11:51:28	Indeterminate	An event with no matching configuration was received from interface 172.20.1.10.
<input type="checkbox"/>	180934	2/4/12 10:51:05	Normal	The Node with Id: 9; ForeignSource: ; ForeignId: has completed.

More...

Recent Outages

There have been no outages on this node in the last 24 hours.

Questions, Contact

Ask away!

identi.ca: [@jeffg](#) / [!opennms](#)

E-mail: jeffg@opennms.org

IRC (Freenode): [jeffg](#),
[#opennms](#)

G+: <http://gplus.to/jeffgdotorg>

FIN

License

This work is licensed under the terms of the Creative Commons Attribution-ShareAlike 3.0 license.

<http://creativecommons.org/licenses/by-sa/3.0/>