

## Foreman - Feature #3506

### Tool to register existing hosts back to Foreman

10/25/2013 11:28 AM - Anonymous

<b>Status:</b> New	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b> Host creation	
<b>Target version:</b>	
<b>Difficulty:</b>	<b>Fixed in Releases:</b>
<b>Triaged:</b>	<b>Found in Releases:</b>
<b>Bugzilla link:</b> 1156488	<b>Red Hat JIRA:</b>
<b>Pull request:</b>	
<b>Description</b> As a user, I would like to provision a machine manually outside of foreman and puppet and then register t back to foreman to be managed by puppet and hostgroups	
<b>Related issues:</b> Related to Hammer CLI - Bug #4358: Unable to create unmanaged host <span style="float: right;"><b>Closed</b>      <b>02/17/2014</b></span>	

#### History

##### #1 - 10/25/2013 11:38 AM - Dominic Cleal

- Status changed from New to Need more information

Foreman can learn about new hosts through Puppet facts and reports already.

##### #2 - 10/29/2013 01:56 PM - Anonymous

- Puppet certificate generation and Puppet registration  
- Create subscription manager plugin to trigger puppet registration

##### #3 - 10/29/2013 02:03 PM - Dominic Cleal

- Subject changed from *As a user, I would like to provision a machine manually outside of foreman and puppet and then register t back to foreman to be managed by puppet and hostgroups* to *Tool to register existing hosts back to Foreman*  
- Description updated

##### #4 - 10/29/2013 02:03 PM - Dominic Cleal

- Status changed from Need more information to New

##### #5 - 02/14/2014 10:43 AM - Lukas Zapletal

- Status changed from New to Assigned  
- Assignee set to Lukas Zapletal  
- Target version set to 1.9.1

What a nice task, taking from backlog.

So basically a rhsm plugin to do what Foreman does during provisioning (register to puppet master).

##### #6 - 02/14/2014 10:51 AM - Dominic Cleal

Yeah, we'd discussed an implementation consisting of a standalone command (maybe even a Puppet face if they weren't going away, or a "puppet-\$foo" git-style subcommand) and then an rhsm plugin can call that. The command would need an API on the Foreman side though for cert signing, as we don't have anything for it today.

##### #7 - 02/14/2014 01:59 PM - Lukas Zapletal

Yeah, this looks feasible:

- new API call that creates autosign entry (if enabled) and returns puppet.conf
- the API action expect a snippet named "puppet.conf" which is rendered and passed to the client
- new package subscription-manager-plugin-foreman
- command puppet-foreman with a --register "subcommand" that calls the API
- rhsm plugin that calls this command with the given hostname

Please comment.

#### #8 - 02/14/2014 02:04 PM - Dominic Cleal

Lukas Zapletal wrote:

Yeah, this looks feasible:

- new API call that creates autosign entry (if enabled) and returns puppet.conf
- the API action expect a snippet named "puppet.conf" which is rendered and passed to the client

You'll need to consider how this is rendered, since you usually have a @host context available and data such as the Puppet and Puppet CA proxies it's assigned to. Will you have to precreate the host, maybe via a facts upload, and/or associate with a host group to pick up this? Or specify the proxy IDs/names you want to use from the CLI?

- new package subscription-manager-plugin-foreman
- command puppet-foreman with a --register "subcommand" that calls the API
- rhsm plugin that calls this command with the given hostname

It'd be good to keep the puppet-foreman bit outside of RHSM as it's useful in other contexts. Maybe this should actually be in hammer-cli-foreman?

Please comment.

Sounds like the right track to me.

#### #9 - 02/14/2014 02:41 PM - Lukas Zapletal

You'll need to consider how this is rendered, since you usually have a @host context available and data such as the Puppet and Puppet CA proxies it's assigned to. Will you have to precreate the host, maybe via a facts upload, and/or associate with a host group to pick up this? Or specify the proxy IDs/names you want to use from the CLI?

Damn you're right, I have no context and I also don't know which proxy. I was hoping to leverage puppet to do the hard work for me (including our future plan of associating org/location when creating new host).

I like the idea of pre-creating host entry (but not via facts upload) and CLI options: org, loc and host group. Rhsm plugins cannot define new options, but we could raise a RFE and for the first version users could define those things in plugin config file. Also the plugin could interactively ask (rhsm does that for credentials already).

It'd be good to keep the puppet-foreman bit outside of RHSM as it's useful in other contexts. Maybe this should actually be in hammer-cli-foreman?

Hmmm hammer will not be usually installed on managed hosts, I was expecting to use just the ruby binding.

#### #10 - 02/14/2014 03:21 PM - Dominic Cleal

Lukas Zapletal wrote:

It'd be good to keep the puppet-foreman bit outside of RHSM as it's useful in other contexts. Maybe this should actually be in hammer-cli-foreman?

Hmmm hammer will not be usually installed on managed hosts, I was expecting to use just the ruby binding.

Yeah, I was thinking it might save some work, or at least building on top of hammer-cli.

#### #11 - 02/14/2014 04:41 PM - Lukas Zapletal

I think I can reuse create host API call, CLI currently has a bug when unmanaged host requires some options. I will fix that first so I can prototype fast and the only required new API would be to get the snippet configuration.

```
[lzap@lzapx hammer-cli]$ be bin/hammer host create --name test --managed false
```

Could not create the host:

```
Error: Options --environment-id, --architecture-id, --domain-id, --puppet-proxy-id, --operatingsystem-id, --partition-table-id are required
```

```
See: 'hammer host create --help'
```

## #12 - 02/17/2014 03:35 PM - Lukas Zapletal

Ok I have fixed the CLI to allow creating of unmanaged hosts and I can confirm this works:

```
hammer host create --managed false --environment-id 1 --name host.any.domain
```

Required information to generate valid puppet configuration:

- Username and Password into Foreman
- Hostname (of the end system trying to register)
- Puppet Environment (ID)
- Puppet CA Proxy name (ID)
- Puppet Master Proxy name (ID)

Optional information

- Organization (ID)
- Location (ID)
- Host group (ID)
- Parameters (keyvalue pairs)

Areas of investigation:

Credentials. I assume this is post-register task, so Foreman API must be integrated with Katello/Candlepin to some degree (client cert-based authentication).

Activation keys. How does activation keys work with this scenario. I assume we want to let users this to be working with AKs. Then we need to extend Katello with all the parameters above, or we can re-use Host Groups for that.

We need to interactively ask for the three puppet things above. If only one Environment, CA or MAster Proxy is found, we can carry on without further questions maybe.

Additional questions. Are these (Org/Loc/Host group) in the scope of this task? Since this is expected to be primarily interactive action, we should offer these.

Template for puppet configuration that needs to be fetched from Foreman:

<https://github.com/theforeman/community-templates/blob/master/snippets/puppet.conf.erb>

To me it currently looks like this architecture:

- puppet-foreman Ruby script utilizing Foreman API gem that does the registration
- rhsm-puppet-foreman-plugin calling the puppet-foreman command post-registration
- API changes on the Foreman side
- RFE for rhsm: allow plugins to define new command line options (for future use)

Opinions? @dominic @bryan

## #13 - 02/17/2014 03:44 PM - Dominic Cleal

Host group may be useful for automation as it can provide defaults for the Puppet proxies and environment.

Location/org are important, but perhaps you can simply hook on the back of [#3214](#) and use the default loc/org settings.

I'd leave Candlepin and AKs out of this, since it's not a Foreman concern.

Sounds good.

## #14 - 02/18/2014 10:16 AM - Lukas Zapletal

Host group may be useful for automation as it can provide defaults for the Puppet proxies and environment.

So we ask user Puppet info OR Host group?

Location/org are important, but perhaps you can simply hook on the back of [#3214](#) and use the default loc/org settings.

Ok

I'd leave Candlepin and AKs out of this, since it's not a Foreman concern.

Hmm but this means we will always ask twice for credentials. Once for Candlepin and once for Foreman.

**#15 - 02/18/2014 10:22 AM - Dominic Cleal**

Lukas Zapletal wrote:

Host group may be useful for automation as it can provide defaults for the Puppet proxies and environment.

So we ask user Puppet info OR Host group?

They're not really exclusive - a host group might provide none, some, or all of the attributes required. I don't know how well, if at all, this works through the host create API though.

**#16 - 02/18/2014 01:18 PM - Lukas Zapletal**

Ok after further investigation, I am putting the rhsm plugin out of scope of our (Foreman team) effort and I will raise new issue on Katello (linking each other) for that.

Goals of this task:

Provide a way for a Foreman user to quickly configure puppet agent using a dedicated script based on hammer CLI client. It will be interactive or via command line options accepting the following options:

- Username and Password
- Hostname (of the end system trying to register)
- Puppet Environment name
- Puppet CA Proxy name
- Puppet Master Proxy name
- Host group name (optional)

The script creates (unmanaged) host entry setting up Puppet parameters and fetching puppet.conf from Foreman and configuring agent to start. The puppet agent then uploads facts.

Artifacts:

- puppet-foreman bash script utilizing Hammer Foreman CLI
- fixing creation of unmanaged hosts via CLI
- new unattended action to render puppet.conf template via token or IP (\*)

(\*) - this might be also implemented as a new Hammer CLI action instead of unattended

**#17 - 02/18/2014 01:31 PM - Lukas Zapletal**

- Related to Bug #4358: Unable to create unmanaged host added

**#18 - 03/03/2014 01:06 PM - Anonymous**

- Target version changed from 1.9.1 to 1.9.0

**#19 - 03/26/2014 01:29 PM - Anonymous**

- Target version changed from 1.9.0 to 1.8.4

**#20 - 04/11/2014 03:11 PM - Lukas Zapletal**

Adding relevant thread, I hope to get back to this soon.

[https://groups.google.com/forum/#!searchin/foreman-dev/devan\\$20goodwin/foreman-dev/B7lfn5lmwSs/jSF6bVyCP1qJ](https://groups.google.com/forum/#!searchin/foreman-dev/devan$20goodwin/foreman-dev/B7lfn5lmwSs/jSF6bVyCP1qJ)

**#21 - 04/24/2014 12:05 PM - Lukas Zapletal**

- Target version changed from 1.8.4 to 1.8.3

**#22 - 05/21/2014 01:21 PM - Anonymous**

- Target version deleted (1.8.3)

**#23 - 10/30/2014 12:43 PM - Dominic Cleal**

- Bugzilla link set to 1156488

**#24 - 10/31/2014 03:06 AM - Marek Hulán**

Lukas, instead of creating some custom bash script, I'd take similar approach as we did in staypuft. We created staypuft-client-installer which is kafo based installer. We use the same puppet module that we have in foreman-installer just a bit different answer file. Wizards for gathering additional information are currently being added to kafo so you could use them. Plus one benefit is that we could use this client installer in configuration templates instead of generating puppet.conf manually and running puppet to reconfigure itself. Maybe we could even server answer files from foreman. Let me know if you need more info.

**#25 - 10/31/2014 05:15 AM - Lukas Zapletal**

Marek, I don't understand you at all. Give me a tinkle :-)

I'd like to work on a design document on this one before making any further steps.

**#26 - 10/31/2014 05:18 AM - Dominic Cleal**

He means a new client-only installer that uses [our puppet module](#), albeit in a client rather than puppet master mode.

**#27 - 10/31/2014 07:54 AM - Lukas Zapletal**

Okay, thanks for the tip. Although the story is: Register the host within Foreman using "subscription-manager register". I don't know how much interactivity can we offer.

In the discussions (above thread link) it looks like the preferred way is to configure stuff in Foreman in advance and only interactively select environment (which rhsm already supports). Therefore there will be probably none interactivity on the client. Need to write some design docs and discuss.

**#28 - 10/31/2014 08:14 AM - Marek Hulán**

Interactivity is optional of course but the same installer would be reused. I was thinking more about the original issue description which doesn't mention subscription manager. But I think we could run the client-installer from subscription manager plugin as well (non interactive mode) while the client-installer could be used elsewhere as well.

**#29 - 10/31/2014 08:21 AM - Stephen Benjamin**

There's an existing puppet module for rhsm - <https://forge.puppetlabs.com/gerapeldoorn/rhsm>, not sure how complete it is, but i can help if needed...

Isn't kafo overkill for this? We have existing finish scripts that do these kinds of things already. I guess it's fine as long as Foreman can support Katello/RHSM too, I really don't want to have to maintain another fork of another kafo installer just for Katello :-(

**#30 - 10/31/2014 08:23 AM - Marek Hulán**

I would say that client-installer package could be generated from foreman-installer (similarly katello-installer) - same codebase and specfile.

**#31 - 11/03/2014 08:55 AM - Lukas Zapletal**

- Status changed from Assigned to New

- Assignee deleted (Lukas Zapletal)

Guys I am busy these days and I am rolling this feature for too much long. Giving up for this one.

If you want to take it, read all the comments here and the thread which contains blueprint how this should work. Feel free to ping me with questions.

**#32 - 11/06/2014 12:39 PM - Stephen Benjamin**

If no one from Foreman picks this up, I can take a look, but I'm not convinced of the need for a Kafo installer, at least not for Katello users. You'd be introducing a chicken-and-egg problem: to run subscription-manager and get yum content, they'd need to install this client package with all its dependencies...

Users would also be forced to install Puppet on clients (e.g. even Salt, Chef, Katello users who may not want it).

I'd say a script would be better, and we may do that over in KatelloLand anyway: a script also has the advantage of being easily wgettable.

**#33 - 12/02/2014 04:23 AM - Stephen Benjamin**

What about adding a "Bootstrap" button to Host groups (if the host group is sufficient to get a finish template back)?

Click the button and you get a drop down window

```
curl -sSL https://foreman.example.com/unattended/bootstrap?token=0906b76d-43e9-40a8-aaca-c8e61eb7459a&hostname=${hostname -f} | bash -s stable
```

The generated token could be used for unlimited hosts within a period of say, 60 minutes (configurable).

When a host fetches that URL, we create the host in Foreman, run the orchestration, and deliver back the relevant finish template.

Because it's host group and finish template based, everything comes for free.