

## Foreman - Refactor #29444

### Improve NIC parsing code

03/31/2020 09:28 AM - Lukas Zapletal

<b>Status:</b>	New	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	Lukas Zapletal	
<b>Category:</b>	Facts	
<b>Target version:</b>		
<b>Difficulty:</b>		<b>Fixed in Releases:</b>
<b>Triaged:</b>	Yes	<b>Found in Releases:</b>
<b>Bugzilla link:</b>	1819751	<b>Red Hat JIRA:</b>
<b>Pull request:</b>		

#### Description

Discussed in the thread <https://community.theforeman.org/t/rfc-drop-parsing-nic-interfaces/17882> but TLDR, we need to:

- make a decision to build on top of facter as the defacto standard for all facts in Foreman
- extend facter 2, 3 and NG with network link information
- define minimum set of facts Foreman understand and document it
- create transformation core API so other fact parsers can easily turn their format into our stanard

Although I was a bit against facter, I think it makes a lot of sense to build on top of it. I think we all agree here.

Extending with network link info (bond, vlans) is already in the works, I have a prototype external facts, currently working on rough edges and want to start on Foreman part soon.

Defining minimum set of facts Foreman understand is the most important one I think. Even facter NG (next generation) is not perfect, I don't like how it throws all facts into one heap - volatile ones (e.g. free memory, uptime) with stable ones (available memory, boot time). It also reports huge amount of facts, we should cherry pick them throwing irrelevant away (these could be enabled by user via settings perhaps - whitelisting mentioned above). We should define minimum set we support (based on facter 3/NG facts plus some extra - e.g. network link we discussed above), we should define which facts are considered stable and which volatile and then refactor our parsing code in a clean way.

A good API for plugin authors is viable to get this into the right direction. Huge advantage is that user is presented with just one set of facts e.g. `networking.interfaces.eth0.bindings6.ipaddress` and this remains the same no matter which client reports this: Ansible, Facter or RHSM.

I created a script that calls facter 2, 3 and 4 (NG) with various NIC configurations (create a VM for that):  
<https://github.com/lzap/facter-nic-tests>

This will give us good enough example JSON outputs to write tests for.

#### History

#1 - 04/01/2020 01:07 PM - Lukas Zapletal

- Bugzilla link set to 1819751