Foreman - Bug #2205

Environment doesn't exist in Puppet 3+ facts YAML

02/13/2013 11:46 AM - Paul O'Connor

Status:	Resolved		
Priority:	Normal		
Assignee:			
Category:	Facts		
Target version:			
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:			

Description

Using Puppet 3 on client nodes results in Foreman setting/overriding the environment set for nodes.

Environment is set properly while using Puppet 2.7

Related issues:

Related to Foreman - Feature #15972: Foreman fact name used for 'update_envir...	**New**	**08/03/2016**

History

#1 - 02/13/2013 11:59 AM - Dominic Cleal

- Subject changed from Environment doesn't exist in Puppet 3+ reports to Environment doesn't exist in Puppet 3+ facts YAML

- Category changed from External Nodes to Facts

- Assignee deleted (Ohad Levy)

Paul's setup has enc_environment = false and update_environment_from_facts = true, so the ENC isn't authoritative over the environment, instead the agents are. In Puppet 3, the facts YAML that's stored on the master and uploaded into Foreman via the ENC node.rb doesn't contain the $environment, while on 2.x it did. This means the environment isn't correctly set on the host in Foreman.

The environment is present in the reports, but:
a) we don't set the host's environment from a report
b) it's too late for the ENC process

Almost certainly a puppet bug or change, but it's possible we should also import environment from reports.

#2 - 06/05/2013 05:23 PM - Anonymous

just a quick "me too"...
could this be added to 1.2? If not, a possible workaround would also be to establish a custom fact "environment" on Puppet 3.x clients.

#3 - 06/05/2013 05:43 PM - Dominic Cleal

- Target version set to 1.3.0

Michael Moll wrote:

> just a quick "me too"...
> could this be added to 1.2? If not, a possible workaround would also be to establish a custom fact "environment" on Puppet 3.x clients.

Probably not, since it would involve adding a new setting I think to control setting environments from reports and new code to handle it (currently in RC phase).

A custom fact sounds like a good plan, provided Puppet doesn't filter ones called "environment" out. It was probably removed because at that point in the agent process, the environment could still be overridden by the puppetmaster (if enc_environment were true, anyway).

#4 - 06/05/2013 06:21 PM - Anonymous

OK, here is my custom fact, which I just verified to work:

```
require 'puppet'

if Facter.value(:puppetversion).to_i >= 3
  Facter.add(:environment) do
    setcode do
      Puppet[:environment]
    end
  end
end
```

#5 - 09/16/2013 12:27 PM - Lukas Zapletal

- Description updated

- Target version changed from 1.3.0 to 1.4.0

#6 - 10/16/2013 03:37 PM - Greg Sutcliffe

- Target version deleted (1.4.0)

#7 - 03/30/2014 11:04 AM - Boyan Tabakov

Are there any plans to implement updating host environment based on reports (and setting to control this)? I think this would be very useful, especially in situations where people use Foreman only for reporting/stats. I was thinking of implementing this, if work hasn't already been starter.

#8 - 03/31/2014 07:20 AM - Dominic Cleal

Nobody's started, I think that'd be appreciated, thanks.

#9 - 12/01/2014 11:29 AM - David Goldsmith

Michael, I'm using Puppet 3.7.3 on CentOS 7.0 and just tried your facter snippet and it did not work. My client is currently in an environment named yum_test (set in [agent] section of /etc/puppet/puppet.conf). My results are:

```
[root@puppetdb21a ~]# puppet agent -t
Info: Retrieving pluginfacts
Info: Retrieving plugin
Notice: /File[/var/lib/puppet/lib/facter/environment.rb]/ensure: defined content as '{md5}ac9d6b73454314a1da95e227eb9fdbbc'
Info: Loading facts
Info: Caching catalog for puppetdb21a.sans.org
Info: Applying configuration version '1417450699'
Notice: Finished catalog run in 2.76 seconds
[root@puppetdb21a ~]# puppet agent --configprint environment
yum_test
[root@puppetdb21a ~]# facter -p environment
production
```

If I replace the "Puppet[:environment]" line with "Facter::Util::Resolution.exec('puppet agent --configprint environment ')" then it works.

But given that "puppet agent --configprint environment" is returning the actual value, where might "Facter[:environment]" be getting "production" from?

#10 - 12/01/2014 11:36 AM - Anonymous

David, I didn't use that fact for a long time now, but still would expect it to work, when used with "puppet agent --test". Could you check that? It probably never worked for the facter -p case.

#11 - 12/02/2014 09:07 AM - David Goldsmith

Ok, I tried again and it does work, mainly.

I am using the theforeman/puppet and the theforeman/foreman modules to setup my servers. The puppet module dynamically rebuilds the puppet.conf file each run if there are any changes, including something like the command line argument "--environment yum_test".   I'm also changing the setting update_environment_from_facts to true in Foreman with the goal of seeing what environment the servers are really in as I am not using Foreman as the ENC and classifying from within.

With your code snippet restored, it did actually update the host status within Foreman and more importantly, when I run "puppet agent -t" on a box in a different environment, it is picking up the new environment's hieradata directory which is the desired behavior. "factor -p environment" still shows "production" on the server even if it is in a different environment, even after multiple puppet agent runs.

When I had tried with my exec statement, it was updating the environment info displayed by "factor -p" but it was also overriding the command-line "--environment" argument which meant I could not dynamically switch environments. I had to edit /etc/puppet/puppet.conf and change the environment there manually, then run puppet and it picked up the "new" environment setting and used the appropriate hieradata.

Given that other facts that I have written show up in the output of "factor -p", why would this one not? (using your snippet)

#12 - 12/02/2014 05:33 PM - Anonymous

That's a Puppet question, as Foreman is out of the whole picture there, but anyway, here we go.. :)

That "require 'puppet'" (which might be too much anyway) in the custom fact's code is loading the necessary code portion from Puppet into facter, but it doesn't do any Puppet agent related initialization, so if you just run "facter -p" it will run with all the defaults from the Puppet code, which is e.g. production for the environment, as the -p switch to my knowledge just extends facter's library path to also include e.g. /var/lib/puppet/lib/facter/ so you can use the facts coming from pluginsync via Puppet.

When running under the Puppet agent, however, facter is spawned under the whole Puppet agent, so all the initialization was already done.

#13 - 08/03/2016 06:53 PM - Robert Heinzmann

- Related to Feature #15972: Foreman fact name used for 'update_environment_from_facts' should be parametrizable added

#14 - 08/14/2016 12:00 PM - Anonymous

As this is a Puppet change, Foreman can nothing do about, I'm leaning towards closing this issue. Opinions?

#15 - 12/30/2016 09:57 AM - Anonymous

- Status changed from New to Resolved

closing, as this is a Puppet issue.