Smart Proxy - Bug #2878

Import Classes fails when using parser = future

08/05/2013 10:36 AM - Tomas Edwardsson

Status:	Closed		
Priority:	Normal		
Assignee:			
Category:	Puppet		
Target version:	1.4.0		
Difficulty:		**Fixed in Releases:**	
Triaged:		**Found in Releases:**	
Bugzilla link:		**Red Hat JIRA:**	
Pull request:			

Description

I have manifests which use the 3.2 experimental features described in http://docs.puppetlabs.com/puppet/3/reference/lang_experimental_3_2.html using **parser=future** in the [master] section.

Version: foreman-proxy-1.2.0-1.el6.noarch

I'm getting the following error in the smart proxy **access.log**

```
Error while parsing /etc/puppet/modules/ok_development/servicedesk/manifests/mail/config.pp: Synta
x error at '.'; expected '}' at line 11
Error while parsing /etc/puppet/modules/ok_development/ssl_cert/manifests/init.pp: Syntax error at
 '.'; expected '}' at line 2
```

Line 11 contains:

```
        $map_emails.each |$index, $value| {
```

One of the manifests that fail looks like this:

```
class servicedesk::mail::config( $apikey, $apiurl, $ignore_senders, $save_all_mails, $map_emails)
{
        file { "/etc/servicedesk/mail.ini":
                owner => root,
                group => mail,
                mode => 640,
                content => template("servicedesk/mail.ini.erb"),
        }

        include mtr

        $map_emails.each |$index, $value| {
                $alias = regsubst($index, '@', '-')
                mtr::sendmail::config::map_entry { $index:
                        recipient => "servicedesk-mail-${alias}",
                        entry_type => 'virtusertable' }
                mailalias { "servicedesk-mail-${alias}":
                        recipient => "| /etc/smrsh/servicedesk-mail ${value}" }
        }
}
```

Associated revisions

Revision af58b9f1 - 01/03/2014 12:06 PM - Rickard von Essen

fixes #2878 - Import Classes when using parser = future, on Puppet 3.2+

History

#1 - 08/05/2013 10:39 AM - Dominic Cleal

- Project changed from Foreman to Smart Proxy

- Category set to Puppet

- Target version set to 1.3.0

#2 - 08/12/2013 08:28 AM - Tomas Edwardsson

Also worth noting is that this module applies cleanly, there are no errors as far as puppet is concerned.

#3 - 08/29/2013 04:47 PM - Aaron Whiteside

Could the fix to this be added as a patch to the 1.2 branch?

#4 - 08/29/2013 04:57 PM - Dominic Cleal

Aaron Whiteside wrote:

> Could the fix to this be added as a patch to the 1.2 branch?

Sorry, no fix has been written for this yet.

#5 - 09/16/2013 12:40 PM - Lukas Zapletal

- Target version deleted (1.3.0)

#6 - 09/23/2013 11:30 AM - Marcus Philip

Hey, this was just removed from 1.3 I see. Isn't this quite serious? There is no known simple workaround, and it's preventing me from using Foreman unless I refactor my whole puppet code base to get rid of lambdas (which would mean also refactoring my hiera data).

Note that even though the future parser function is labeled 'experimental', it's most likely going to be the parser used in puppet 4.0, for which I know of no release date, but I'm guessing around new year, judging from their current release cadence.

Puppet labs state that 'features in the experimental parser are exempt from semantic versioning', but I don't understand why the foreman proxy really has to care. I don't know the internal design of the proxies, but isn't it the puppet master that compiles the puppet manifests. Why can't just Foreman (and/or the proxy) accept the compiled manifest?

Does anyone know of a monkey patch that fixes this?

Can someone indicate how I would solve this so I can send a pull request?

Refs:
http://docs.puppetlabs.com/puppet/3/reference/release_notes.html#experimental-future-parser-with-iteration
http://docs.puppetlabs.com/puppet/3/reference/lang_experimental_3_2.html

#7 - 09/23/2013 11:46 AM - Dominic Cleal

The removal from 1.3 just means that nobody has produced a patch in time, while 1.3's scheduled release date was reached. Sorry.

I agree it's quite important, but I don't know what needs to change to make it work. The proxy actually calls Puppet's parser and then runs over the AST generated to find out about classes and the parameters they take. I don't know how much of the future parser is shared with the old one, so it could just be a case of sending a switch to enable the future parser, or it could have a significant effect on how we use the AST today. If you want to start looking into it, lib/proxy/puppet/puppet_class.rb is the starting point.

#8 - 09/23/2013 12:06 PM - Marcus Philip

Dominic Cleal wrote:

> The removal from 1.3 just means that nobody has produced a patch in time, while 1.3's scheduled release date was reached. Sorry.

Ok. Great that you're keeping the issue tracker tidy and uip to date.

> The proxy actually calls Puppet's parser and then runs over the AST generated to find out about classes and the parameters they take. I don't know how much of the future parser is shared with the old one, so it could just be a case of sending a switch to enable the future parser, or it could have a significant effect on how we use the AST today.

As I said, I'm surprised it doesn't just work. I think the output is identical, as it works for all my classes that doesn't have lambdas. It seems the Smart-Proxy doesn't get the *future* parser even though I have parser = future in both [main] and [master] section of puppet.conf.

> If you want to start looking into it, lib/proxy/puppet/puppet_class.rb is the starting point.

Thanks. I'm looking at it... :)

#9 - 09/23/2013 12:14 PM - Dominic Cleal

Marcus Philip wrote:

> Dominic Cleal wrote:
>
> > The proxy actually calls Puppet's parser and then runs over the AST generated to find out about classes and the parameters they take. I don't know how much of the future parser is shared with the old one, so it could just be a case of sending a switch to enable the future parser, or it could have a significant effect on how we use the AST today.
>
> As I said, I'm surprised it doesn't just work. I think the output is identical, as it works for all my classes that doesn't have lambdas. It seems the Smart-Proxy doesn't get the *future* parser even though I have parser = future in both [main] and [master] section of puppet.conf.

Agreed, this to me suggests that we're hooking at too low a level, so always hitting the old parser instead of taking the Puppet configuration into account. The other possibility is that we're not loading the Puppet config at all when initialising the parser, I'm unsure.

> > If you want to start looking into it, lib/proxy/puppet/puppet_class.rb is the starting point.
>
> Thanks. I'm looking at it... :)

Cheers!

#10 - 09/23/2013 01:16 PM - Marcus Philip

I think the problem is that Proxy::Puppet::PuppetClass is doing Puppet::Parser::Parser.new and not using the Puppet::Parser::ParserFactory.

I think I have done a fix but I see no effect. Possibly I do not know enough Ruby to understand how to deploy my change. I tried changing: /usr/share/foreman-proxy/lib/proxy/puppet/initializer.rb directly on my puppet master, but see same error when I click 'import puppet classes' in foreman. Isn't this the file that is used?

#11 - 09/23/2013 01:18 PM - Dominic Cleal

Yes, that's the file. You'd want to restart foreman-proxy after editing it.

#12 - 09/23/2013 02:32 PM - Marcus Philip

Thanks for the help. I thought the proxy was running in httpd like foreman itself...

That did indeed make my change take effect, but unfortunately it seems maybe you are right: The interface doesn't seem to be the same. I get lots of error like this:

> Error while parsing /etc/puppet/environments/dev/modules/soapui/manifests/init.pp: undefined method `known_resource_types' for #<Puppet::Parser::EParserAdapter:0x7f6d1fb2cdf0>

I don't understand why.

#13 - 09/24/2013 08:38 AM - Marcus Philip

Marcus Philip wrote:

> I tried changing: /usr/share/foreman-proxy/lib/proxy/puppet/initializer.rb

I mean of course /usr/share/foreman-proxy/lib/proxy/puppet/puppet_class.rb

#14 - 09/29/2013 08:48 PM - Henrik Lindberg

It is correct that the EParserAdapter is not 100% compatible with the original parser. Some of the convenience methods available on the original parser are not included in the new.
This is especially true for anything that requires knowledge about the environment (which the parser does not need).

Code that needs `known_resource_types` should obtain those from the environment (it is implemented there). The environment can be obtained from `scope.environment`, `compiler.environment`, or `node.environment` (to get the environment associated with the particular object), or `Puppet::Node::Environment.current` to get the environment in effect.

Also note that the current implementation (in Puppet 3.x) first parses to a new AST model, and then transforms this to the 3.x regular AST before evaluating it. This will change going forward as the new AST model will be evaluated directly by a new evaluator. The new AST model is not compatible with the old. When this switch takes place, and as long as the 3x AST is kept around it will be possible to transform to it (simple - it is one call). At some point code that uses the 3x AST will need to be modified.

It is not unreasonable to request that an "info" API is written that returns information about "what was parsed" in an AST model version neutral way.

#15 - 12/22/2013 10:50 AM - Rickard von Essen

I have made a patch for this, see https://github.com/theforeman/smart-proxy/pull/123 . If some one have lots of puppet manifests with uses new constructs from the future parser *testing is welcomed*.

#16 - 01/02/2014 02:59 PM - Dominic Cleal

- Status changed from New to Ready For Testing

- Target version set to 1.9.3

#17 - 01/03/2014 12:08 PM - Dominic Cleal

- translation missing: en.field_release set to 2

#18 - 01/03/2014 12:52 PM - Rickard von Essen

- Status changed from Ready For Testing to Closed

- % Done changed from 0 to 100

Applied in changeset af58b9f1ee7fe84ed1d94e36ef744a2d711c3b33.

#19 - 01/03/2014 12:56 PM - Dominic Cleal

This will be available in the nightly packages very shortly, and the proxy is backwards-compatible with the current release(s) of Foreman. If anybody's able to test this, please install foreman-proxy from the nightlies and see how it works (file new bugs if necessary), else it'll be available in 1.4.0 RC1 soon.