

Foreman - Feature #36

Clicking of classes in create group/node should determine the order

10/06/2009 09:50 AM - Matt Moran

Status: Rejected	
Priority: Normal	
Assignee:	
Category:	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
When adding classes to a group or node, the click of adding should determine the order the classes are added.	
As there are many if tagged {} operations in our code, I have to add a class click update, add the next etc. to determine the correct order.	

History

#1 - 10/06/2009 05:49 PM - Ohad Levy

what are the use cases for those tags?

as its very specific to your infrastructure, maybe you can create your own class which include the classes in the right order?

#2 - 10/07/2009 09:14 AM - Matt Moran

I don't think it's specific to our infrastructure as many people on the mailing list use it. Although I do think it is more of a puppet issue i.e. puppet should collect all tags at the start of the catalogue so it can process the 'if tagged("webserver")' syntax.

I also think the idea of having a class which is a node is a bit of a overhead. Most nodes do only include a handful of classes, but there are many times especially in test/dev environments when you require one node to be many things. This means for each possible configuration of a node, a 'dummy' class will need to be created. I think this is more of a link to groups in foreman.

#3 - 10/07/2009 05:55 PM - Ohad Levy

Matt Moran wrote:

I don't think it's specific to our infrastructure as many people on the mailing list use it. Although I do think it is more of a puppet issue i.e. puppet should collect all tags at the start of the catalogue so it can process the 'if tagged("webserver")' syntax.

this is exactly why I started to use external nodes, using the host parameters, you can define something which will always be correct to the running host, regardless of the manifest order.

for example, you could setup default variables (in variuos levels) and override them in the domain/hostgroup/host level. this should allow you to get rid of the if tags syntax.

I also think the idea of having a class which is a node is a bit of a overhead. Most nodes do only include a handful of classes, but there are many times especially in test/dev environments when you require one node to be many things. This means for each possible configuration of a node, a 'dummy' class will need to be created. I think this is more of a link to groups in foreman.

if you use foreman, you have a very powerful build system, I would suggest that you test each one of your hosts configuration at a time (as this is what your host will actually be using), and then you dont need to use different test scenarios just for tests.

#4 - 10/08/2009 12:26 AM - Matt Moran

Very true about replacing external nodes with if tagged() syntax. I just didn't want to get to the stage where there are hundreds of defined external mostly static variables, or are not even relevant to a node, but are there because they are domain specific. It causes a unnecessary verbose yaml output IMHO.

Although there is the benefit of changing these variables outside of the manifest code when using external variables.

Great product by the way Ohad.

#5 - 10/08/2009 12:54 AM - Ohad Levy

Matt Moran wrote:

Very true about replacing external nodes with `if tagged()` syntax. I just didn't want to get to the stage where there are hundreds of defined external mostly static variables, or are not even relevant to a node, but are there because they are domain specific. It causes a unnecessary verbose yaml output IMHO.

why don't you define them in the domain level than?

#6 - 10/08/2009 01:00 AM - Matt Moran

Wouldn't they all still appear in the yaml output, even though they won't be needed for the majority of classes that yaml output shows for that node?

#7 - 10/08/2009 01:03 AM - Ohad Levy

Matt Moran wrote:

Wouldn't they all still appear in the yaml output, even though they won't be needed for the majority of classes that yaml output shows for that node?

That's true, the parameters that are provided via external node are available to all classes (that's why you don't have the problems with scoping). if you want them to appear only to a subset of your hosts, you can either use host-groups parameters or define them per host.

#8 - 10/16/2009 01:26 PM - Ohad Levy

- *Status changed from New to Rejected*

I don't see much sense in implementing this, please reopen if you think its still required.