

Smart Proxy - Bug #4390

Dynamic environments broken

02/19/2014 03:30 PM - Nathan Flynn

Status: Resolved	
Priority: Normal	
Assignee:	
Category: Puppet	
Target version:	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
Hello,	
We use puppet dynamic environments in our platform to make changes without effecting the entire platform.	
We put our 'shared/common' modules in /etc/puppet/modules and copies of some of these modules inside the environment to do dev/test.	
Looking at the Foreman code for dynamic envs it should do the following, look in /etc/puppet/environments/\$environment/modules add all classes then look in /etc/puppet/modules and add any additional classes. What appears to happen is it only looks in /etc/puppet/environments/\$environment/modules and ignores /etc/puppet/modules	
Dynamic set: [master] # Dynamic environments modulepath = /etc/puppet/environments/\$environment/modules:/etc/puppet/modules	
If we use static envs as below it appears to work as expected and pull classes from both paths.	
[dev] modulepath = /etc/puppet/environments/dev/modules:/etc/puppet/modules	
-	
This is currently causing us major issues as foreman is removing classes what the puppetmaster would accept believing they do not existing in the dynamic env.	
ii foreman	1.4.0-1 amd64 Systems management web interface
ii foreman-proxy	1.4.0-1 all RESTful proxies for DNS, DHCP, TFTP, and Puppet

History

#1 - 09/18/2014 03:21 AM - Ohad Levy

- Status changed from New to Feedback

- Priority changed from High to Normal

I assume this is already resolved, can you please confirm?

thanks

#2 - 09/18/2014 04:00 AM - Nathan Flynn

With the recent changes to puppet environments it invalidates this fault.

I don't see a way to close the ticket.

#3 - 09/18/2014 05:00 AM - Dominic Cleal

- Status changed from Feedback to Resolved

Thanks for confirming. The new dynamic environments mode makes our implementation more reliable as we simply call out to Puppet to get the env list, instead of trying to replicate its logic.