

## Foreman - Feature #1959

### Instantiate Puppet define()s from ENC

11/20/2012 03:26 AM - Jan Doleschal

<b>Status:</b> Feedback	
<b>Priority:</b> Normal	
<b>Assignee:</b>	
<b>Category:</b> Puppet integration	
<b>Target version:</b>	
<b>Difficulty:</b>	<b>Fixed in Releases:</b>
<b>Triaged:</b>	<b>Found in Releases:</b>
<b>Bugzilla link:</b>	<b>Red Hat JIRA:</b>
<b>Pull request:</b>	
<b>Description</b> It would be very great if we could reuse parameterized classes. We are using puppet to install some tomcat instances on one machine. It should look like: class tomcat tomcat::tomcat_instance {instance_1 port => 10080 }  class tomcat tomcat::tomcat_instance {instance_2 port => 10080 }	

#### History

##### #1 - 11/20/2012 03:27 AM - Ohad Levy

- Status changed from New to Need more information

I'm not sure I follow, smart variables do exactly that, can you provide a bit more example of how you use it?

##### #2 - 11/20/2012 03:34 AM - Jan Doleschal

Ok, here is better example. We use this class to cleanup some systems. Cut from our site.pp I could not figure out how to realize it with smart variables.

```
node server_1{
```

```
cleanup_2::cleanup {"/var/log/":  
  matches => ["zmd-messages.log.*", "zypper.log-*", "net-snmpd*", "warn*", "zmd-backend", "mail*", "log.*.old", "log.winbind-*.bz2", "*.bz2", "*.bz2-*", "*.log"],  
  recurse => true,  
  size => "100m",  
  age => "4w",  
  rmdirs => false,  
}
```

```
cleanup_2::cleanup {"/var/puppet/clientbucket":  
  matches => "*",  
  recurse => true,  
  size => "50m",  
  age => "12w",  
  rmdirs => true,  
}  
include cleanup_2  
}
```

##### #3 - 11/20/2012 04:10 AM - Ohad Levy

which version of foreman are you using? my guess is that you are using a version before parameterize class support was introduced?

#### #4 - 11/20/2012 04:28 AM - Dominic Cleal

Jan - is the cleanup\_2::cleanup actually a define() rather than a parameterised class? Defines can be used multiple times, so that's what it looks like to me, while a class (even parameterised) can only be included once. If you could paste the start of the cleanup declaration that'd be useful to clarify.

#### #5 - 11/20/2012 05:16 AM - Jan Doleschal

Yes, it is a define. Of course you can include a class only one times. Sorry for the bad explanation. Here is the complete class.

```
class cleanup_2 {
  define cleanup ($matches, $age, $size, $recurse, $rmdirs) {
    tidy {"$name":
      matches => $matches,
      age     => $age,
      size   => $size,
      recurse => $recurse,
      backup => false,
      rmdirs => $rmdirs;
    }
  }
}
```

#### #6 - 11/20/2012 05:30 AM - Dominic Cleal

- Subject changed from Reuse parameterized classes to Instantiate Puppet define()s from ENC
- Status changed from Need more information to Assigned

Ok, thanks - that makes more sense.

I can see the use in this, but Puppet's ENC interface can only add classes to a node and not instantiate resources (or defines). This would depend on an extension to their ENC interface, or for Foreman to create classes of its own with the resources inside, then add that generated class to the node.

#### #7 - 07/22/2014 12:04 PM - Dominic Cleal

- Description updated
- Category set to Puppet integration
- Status changed from Assigned to New

#### #8 - 10/02/2014 11:01 AM - Social Point

Suggestion:

As said on [http://projects.theforeman.org/projects/foreman/wiki/Instantiate\\_Puppet\\_resources](http://projects.theforeman.org/projects/foreman/wiki/Instantiate_Puppet_resources)

We can create a module for all wrappers

then for every define we want we just need to create a manifest, always with the same content.

For example, to be able to instantiate bind::server::conf of this module: <https://forge.puppetlabs.com/thias/bind>

We can create /etc/puppet/environments/production/modules/wrappers/manifests/bind/server/conf.pp with this content:

```
class wrappers::bind::server::conf($resources = {}) {
  create_resources('bind::server::conf', $resources)
}
```

Then on foreman we can include that new class and for resource parameter we can provide this hash:

```
/etc/named.conf:
listen_on_addr:
- any
listen_on_v6_addr:
- any
forwarders:
- 8.8.8.8
- 8.8.4.4
```

#### #9 - 05/20/2017 04:42 AM - Anonymous

- Status changed from New to Feedback

I'm leaning towards closing this, as I can't image that Puppetlabs extend the ENC interface that way in the future