

## Pulp - Story #5662

### Manage Pulp via Ansible modules

11/04/2019 05:15 PM - Anonymous

<b>Status:</b>	CLOSED - CURRENTRELEASE	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0:00 hour
<b>Sprint/Milestone:</b>		<b>Tags:</b>	
<b>Platform Release:</b>		<b>Sprint:</b>	
<b>Groomed:</b>	No	<b>Quarter:</b>	
<b>Sprint Candidate:</b>	No		

#### Description

I've been testing ansible modules to manage Pulp.

<https://github.com/ansible/ansible/pull/64388>

Please let me know what you think.

#### History

##### #1 - 11/04/2019 09:03 PM - bmbouter

- Sprint/Milestone deleted (3.0.0)

##### #2 - 11/07/2019 05:05 PM - bmbouter

@Timoses there is another developer [mdellweg](#) (aka x9c4 on IRC) who is also developing a similar thing. I wanted to connect you both to de-duplicate effort and share testing. If I've misread the similarity then please continue!

##### #3 - 11/07/2019 06:59 PM - mdellweg

@timoses For completeness sake, here's the link to that project [0].

I have already seen, that you took somehow similar but still different approaches to deal with the api. And you seem to be aiming for less modules that can handle e.g. remotes of different types, while i would rather implement one module for each type of remote. I think that way it is easier to account for the subtle specialities, but i'd like to also hear your reasoning.

[0] [https://github.com/mdellweg/ansible\\_modules\\_pulp](https://github.com/mdellweg/ansible_modules_pulp)

##### #4 - 11/15/2019 04:41 PM - Anonymous

[mdellweg](#)

Looks great! Your approach looks a bit cleaner.

What would you think of a `PulpApiLoader` class or similar which would be responsible for loading the APIs and handing the correct classes. All in all I believe that part requires some more abstraction (one function per plugin type and remote/publication/... seems unnecessary since OpenAPI appears to follow strict naming conventions).

If you don't mind I'd try to find the time in the upcoming week to implement something like that to make it easier to add plugin types.

I'd also suggest to take your code base as a basis and combine our efforts. What's your take on it?

Separating modules by plugin type does make the documentation and implementation easier I believe. In my implementation, I like and dislike the dynamic loading of the appropriate module `arg_specs` for various pulp plugins. It's both convenient but at the same time complicates implementation details. Thinking about it now I'd rather go with separate modules just to keep it simple.

From a user perspective I suppose it would not make much of a difference. What do you think?

##### #5 - 10/09/2020 03:39 PM - dkliban@redhat.com

- Status changed from NEW to CLOSED - CURRENTRELEASE

squeezer collection on ansible galaxy.