

Pulp - Task #8204

Add method to allow import of models with references to repo versions

02/04/2021 12:46 AM - gerrod

Status: NEW	Start date:
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category:	Estimated time: 0:00 hour
Sprint/Milestone:	Tags:
Platform Release:	Sprint:
Groomed: No	Quarter:
Sprint Candidate: No	
Description	
Currently pulp import creates all the content models before creating the new repository version meaning that models with foreign keys to repository version can not be imported. This is currently the case with the AnsibleCollectionDeprecated model in pulp_ansible. It's possible other plugins have similarly difficult models to export/import that would need this solution.	
Related issues:	
Blocks Ansible Plugin - Story #8205: Add AnsibleCollectionDeprecated resource...	MODIFIED

History

#1 - 02/04/2021 12:48 AM - gerrod

- Blocks Story #8205: Add AnsibleCollectionDeprecated resource to export/import added

#2 - 02/04/2021 02:24 AM - daviddavis

- Sprint/Milestone set to 3.11.0

#3 - 02/23/2021 07:31 PM - ggaaney

I'm not sure the realities of import/export allow us to do anything useful here.

'downstream' is not (ever) a copy of upstream. It's its own version of Pulp, whose admin can be doing anything they want to repositories (like adding and removing repo-versions). An upstream AnsibleCollectionDeprecated is deprecated for a specific repo-version; how do we map that to a downstream that may, for example, have never had a repo-version? Or one that has had 100, when upstream had 3? How do we expect the AnsibleCollectionDeprecatedModelResource to 'find' the repo-version it is deprecated for/against?

[gerroddaviddavis](#), can either of you find any way for us to be able to make this possible? This may be a permanent restriction for this data-flow.

#4 - 02/23/2021 07:39 PM - daviddavis

ggaaney wrote:

I'm not sure the realities of import/export allow us to do anything useful here.

'downstream' is not (ever) a copy of upstream. It's its own version of Pulp, whose admin can be doing anything they want to repositories (like adding and removing repo-versions). An upstream AnsibleCollectionDeprecated is deprecated for a specific repo-version; how do we map that to a downstream that may, for example, have never had a repo-version? Or one that has had 100, when upstream had 3? How do we expect the AnsibleCollectionDeprecatedModelResource to 'find' the repo-version it is deprecated for/against?

We have a specific repo version upstream that has been exported and we're mapping it to the new repo version downstream that is created during import. This newly created repo version could be passed to the plugin.

[gerroddaviddavis](#), can either of you find any way for us to be able to make this possible? This may be a permanent restriction for this data-flow.

One naive solution would be to simply dump a list of collections that should be deprecated. After the new repo version is imported, we create a AnsibleCollectionDeprecated for each collection in the list and attach it to the new repo version.

#5 - 02/23/2021 07:52 PM - daviddavis

Here's maybe a bit better solution: during export AnsibleCollectionDeprecatedModelResource dumps the natural key for collections (namespace, name).

During import, the pulpcore code calls some list of model resources (POST_IMPORT_ORDER or something?) after creating the repo version and it sets a repo version property on these resources and then runs resource.import_data. The resource would have to define before_import_row to mung each row to account for repo version it has.

#6 - 02/23/2021 09:13 PM - ggaaney

OK, I see where my confusion was. Looking at the data model, I assumed that an AnsibleCollectionDeprecated must be associated with an arbitrarily-previous repo-version. The reality is, it is to be associated with, only, the repo-version that is being created at import-time.

Your proposal in #c5 is the direction I'm leaning. It needs some more think-time, but the general path (import post-RV-creation, BaseContentResource gets a resource-version-UUID field, subclasses use before_import_row to take advantage of that if desired) is I think the right direction to take.

Thanks for the clarification!

#7 - 03/02/2021 04:06 PM - ipanova@redhat.com

- *Sprint/Milestone changed from 3.11.0 to 3.12.0*

#8 - 03/19/2021 03:55 PM - mdellweg

- *Sprint/Milestone deleted (3.12.0)*