Pulp - Story #4832

As a user, I can filter all repo versions by content unit

05/16/2019 04:41 PM - daviddavis

Status:	NEW	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Sprint/Milestone:			
Platform Release:		**QA Contact:**	
Blocks Release:		**Complexity:**	
Backwards Incompatible:	No	**Smash Test:**	
Groomed:	No	**Verified:**	No
Sprint Candidate:	No	**Verification Required:**	No
Tags:		**Sprint:**	

Description

If you want to delete a content unit, there's no way to check which repo versions it belongs to. This means that deleting content is virtually impossible.

We need to give users a way to filter repo versions by content. The problem is that there is no global repo version endpoint--repo versions are nested under repos.

Related issues:

Related to Pulp - Story #4831: As a user, I have docs on how to delete content	**NEW**

History

#1 - 05/16/2019 04:42 PM - daviddavis

- Related to Story #4831: As a user, I have docs on how to delete content added

#2 - 05/16/2019 05:21 PM - daviddavis

- Subject changed from As a user, I can filter repo versions by content unit to As a user, I can filter all repo versions by content unit

Looks like there's a repo version filter already for filtering by content:

https://github.com/pulp/pulpcore/blob/master/pulpcore/app/viewsets/repository.py#L96

The only problem is that this is nested under repos.

#3 - 05/17/2019 05:11 PM - amacdona@redhat.com

I am not very fond of the idea that users will need to "delete history" and remove all versions that contained a particular content unit jsut to remove it. As an alternative, I think it would be nice to add a boolean field to content like `disable_distribution`, which will allow the content to be added/removed from repository versions, but will not be served by the content app. Perhaps disabled content should be omitted from publications as well on a plugin-to-plugin basis.

Heres an example of why this is preferable. Say repo x is published and distributed in production, and one of the packages is determined to have a major security flaw. This package has been in the repository since the beginning. Is the user now expected to completely wipe all their history just to prevent the content from being distributed?

#4 - 05/17/2019 05:28 PM - daviddavis

@asmacdo, I think your point makes sense. Let me give another example though: a user uploaded a package with sensitive information like customer info or a password. Even git lets you rewrite history to remove stuff. It's painful but possible.

Also, this story isn't entirely tied to deletion. What if a user wants to see how many/which repo versions have a particular content unit? Like in the case they're doing a security audit or something.

#5 - 05/17/2019 05:29 PM - amacdona@redhat.com

+1 this story is necessary even outside of the content unit deletion use case.

#6 - 05/17/2019 05:37 PM - bmbouter

I think there are several use cases here and we probably need a variety of mechanisms to cover them all.

1. A security-unsafe package is in Pulp and must be removed. We gotta get it out, quickly, and no matter what.
2. Which repo versions contain unit X?
3. Where is unit x being served right now?

What other aspects of this area could we consider use cases? What about ^ use cases, do they make any sense?

#7 - 09/13/2019 05:39 PM - daviddavis

- Sprint/Milestone deleted (3.0)