

Pulp - Story #3209

As a user, I have Repository Versions

12/15/2017 10:11 PM - dkliban@redhat.com

Status:	CLOSED - CURRENTRELEASE	Start date:	
Priority:	Normal	Due date:	
Assignee:	amacdona@redhat.com	% Done:	100%
Category:		Estimated time:	0:00 hour
Sprint/Milestone:	3.0.0	Tags:	
Platform Release:		Sprint:	
Groomed:	No	Quarter:	
Sprint Candidate:	No		

Description

This is the parent task for all stories/tasks related to adding RepositoryVersions.

Repository Version CRUD

CREATE:

New Repository Version can be created in 2 ways, (1) directly with the Repository Versions Viewset (3234) and (2) action endpoints defined by each plugin (see "actions" below).

Manual add/remove is done by a POST to `/v3/repositories/<repo_id>/versions/`. The user specifies the content to add and the content to remove in the POST body. The RepositoryVersions Viewset will deploy the task that creates a RepositoryVersion.

Stories:

<https://pulp.plan.io/issues/3234> Create a new Repository Version
`/v3/repositories/<repo_uuid>/versions/` POST

LIST:

list (`/v3/repositories/<repo_uuid>/versions/`) and read (`/v3/repositories/<repo_uuid>/versions/<number>/`) is handled by the Repository Version serializer and provides content href, added href, and removed href.

Stories:

<https://pulp.plan.io/issues/3218> Add detail routes to viewset and serializer
`/v3/repositories/<repo_uuid>/versions/<number>/` should include links to detail routes, including:

- content href
- added content href
- removed content href

<https://pulp.plan.io/issues/3235> On Repositories Serializer, return href for latest version

<https://pulp.plan.io/issues/3238> Filters for RepositoryVersions

Bugs:

<https://pulp.plan.io/issues/3230> - bugfix - `/v3/repositories/<deleted_repo>/versions/` should return 404

UPDATE: disabled

RepositoryVersions are immutable, so the Viewset should not inherit from the update mixin, so PUT and PATCH requests return a 405 (Method not allowed)

DELETE:

All versions are able to be deleted. The latest version is simply deleted and the latest version becomes the previous latest. When a version other than latest is deleted, the changes made by that version are "squashed" into the following version.

For example, Repository Av1 contains 2 packages (a and b). Av2 Adds version package c. Av3 removes package a. Av4 removes

package b.
Av1 (a, b)
Av2 (a, b, c) [+c]
Av3 (b, c) [-a]
Av4 (c) [-b]

If Av2 is deleted, Av3 now also adds "c". Av3(b, c) [+c, -a]
If Av4 is deleted, A.latest() become Av3 and contains (b, c)

When a user deletes a repository version, the content contained by all following versions does not change, but the content added and removed in the following version changes.

When the latest version is deleted, Repository.latest rolls back to the previous version.

Stories:

<https://pulp.plan.io/issues/3219> Delete Repository Versions:

RepositoryVersion.latest is deleted

RepositoryVersion that is not latest is "squashed"

Bugs

<https://pulp.plan.io/issues/3233> Squash bug

Actions that create a Repository Version (including sync)

If Pulp performs an action to create a Repository Version (as opposed to a manual add/remove), all of the logic is owned by each plugin. Interactions with the pulpcore database (RepositoryVersion, Repository, RepositoryContent) are handled by pulpcore utilities that are part of the plugin API. The utilities will provide a working directory, ensure that writes happen in transactions, clean up pulpcore database after failures and crashes.

This change alters the plugin API. "sync", "publish" and other actions will be tasks that are written by each plugin (instead of a function on the model). The tasks are deployed by Plugin Viewsets which create action endpoints in the REST API.

Transactions and Cleanup stories:

<https://pulp.plan.io/issues/3222> Add complete flag, and mark complete when action is finished.

<https://pulp.plan.io/issues/3226> Cleanup incomplete Repository Versions after a worker crash

<https://pulp.plan.io/issues/3225> Update Changeset with new arguments (repo_version, not old_version and new_version)

Plugin API changes:

<https://pulp.plan.io/issues/3074> Remove sync code from pulpcore. Also acts as parent task for Plugin Stories:

<https://pulp.plan.io/issues/3285> Provide context manager to handle RepositoryVersion creation, finalization, and cleanup.

Plugin Stories

<https://pulp.plan.io/issues/3260> File plugin sync

<https://pulp.plan.io/issues/3294> - Python plugin sync

Actions that create a Publication (including publish)

Actions that create Publications are handled similarly to actions the create RepositoryVersions. All of the logic is owned by each plugin. Interactions with the pulpcore database (Publication, Repository) are handled by pulpcore utilities that are part of the plugin API.

Stories:

<https://pulp.plan.io/issues/3221> Move publish code from model to task

<https://pulp.plan.io/issues/3295> Write util to assist with publish (context manager?)

Plugin stories:

<https://pulp.plan.io/issues/3296> file publish story

Docs for Plugin Writer:

<https://pulp.plan.io/issues/3220> Plugin writer HOWTO guide to implement sync (and copy)

<https://pulp.plan.io/issues/3298> Plugin writer HOWTO guide to implement publish

Plugin Docs:

<https://pulp.plan.io/issues/3249> Update file plugin docs for repository versions

Why this design?

Motivation

The ContentUnits added to a repository change over time, but Pulp only stores the current state. This prevents a lot of great things like rolling back to a particular snapshot, or understanding exactly what content is in place at any given time.

Solution

Introducing the Repository Version, which represents the ContentUnits in a repository after any given operation. For example, each of the following operations will produce a repository version.

- Sync with an importer
- Add a set units to a repository
- Remove a set of units from a repository

Any repository version can be deleted. Publications are changed to publish a specific RepositoryVersion, not just the latest content in a given repo.

API examples

```
POST /api/v3/importers/{type}/{importer_id}/sync/ Use plugin to create a new RepositoryVersion.
POST /api/v3/repositories/{repo_id}/versions/ # create the new RepositoryVersion. specify units to
o add, and/or units to remove
GET /api/v3/repository/{repo_id}/versions/{number} # The resource for a single repo version
GET /api/v3/repositories/{repo_id}/versions/<id>/content/ # paginated list of all content in part
icular version
GET /api/v3/repositories/{repo_id}/versions/<id>/added_content/ # paginated list of content added
when a particular version was created
GET /api/v3/repositories/{repo_id}/versions/<id>/removed_content/ # paginated list of content rem
oved when a particular version was created
```

Technical details

The *RepositoryVersion* table stores all repository versions for all repositories. A repository version is a set of content.

RepositoryVersion

- [pk] **id** - The primary key.
- [fk] **content** - The associated repository.
- **created** - When the repository version was created.
- **number** - A positive integer that uniquely identifies a version of a specific repository. Each new version for a repository should have this field set to 1 + the most recent version.
- **complete** - When true, the action that creates the RepositoryVersion is finished and the version is ready to be used.

The *RepositoryContent* table stores association between a repository version and its contained content.

RepositoryContent

- [pk] **id** - The primary key.
- [fk] **content** - The associated content.
- [fk] **repository** - The associated repository.
- [fk] **version_added** - The RepositoryVersion which added the referenced Content.
- [fk] **version_removed** - The RepositoryVersion which removed the referenced Content.

Subtasks:

Refactor # 3074: Move sync logic from models to tasks

Task # 3184: Add complete field to Publication.

Story # 3186: Rewrite tasking system labels for versioned repositories

Story # 3218: As an authenticated user, I can list repository version content, added co...

CLOSED - CURRENT

CLOSED - CURRENT

CLOSED - CURRENT

CLOSED - CURRENT

Story # 3219: As an authenticated user, I can delete any repository version.	CLOSED - CURRENT
Task # 3220: Expand plugin writer's guide to include RepositoryVersion creating actions...	CLOSED - CURRENT
Task # 3221: Move publish logic from models to tasks	CLOSED - CURRENT
Task # 3222: Add 'complete' attribute to the RepositoryVersion model	CLOSED - CURRENT
Task # 3225: Update Changeset API to use repository versions	CLOSED - CURRENT
Story # 3226: As a user, I am guaranteed that my incomplete repository versions are cle...	CLOSED - CURRENT
Issue # 3230: Repository version REST API returns 200 instead of 404	CLOSED - CURRENT
Issue # 3233: Version incorrectly says that it removed a content unit	CLOSED - CURRENT
Story # 3234: As an authenticated user, I can create a new version by adding or removin...	CLOSED - CURRENT
Story # 3235: As an authenticated user, when viewing a repository, I can discover a hre...	CLOSED - CURRENT
Story # 3238: As an authenticated user, I can filter repository versions	CLOSED - CURRENT
Task # 3249: Update pulp_file README	CLOSED - CURRENT
Refactor # 3260: Move sync logic from Importer model to a task	CLOSED - CURRENT
Story # 3274: As a user, the task created by sync includes the RepositoryVersion create...	CLOSED - CURRENT
Story # 3285: As a plugin writer, I have a tool that helps me write tasks that create R...	CLOSED - CURRENT
Task # 3294: Update Python plugin to sync with versioned repositories	MODIFIED
Story # 3295: As a plugin writer, I have a tool that helps me write tasks that create P...	CLOSED - CURRENT
Refactor # 3296: Move publish logic from Publisher model to a task	CLOSED - CURRENT
Task # 3298: Expand plugin writer's guide to include Publication creating actions (incl...	CLOSED - CURRENT
Task # 3300: Update plugin docs after Versioned Repo changes are merged	CLOSED - COMPLETE
Story # 3312: As a user, Publications show me their RepositoryVersion	CLOSED - CURRENT
Story # 3324: As a user, I can publish any repository version	CLOSED - CURRENT

History

#1 - 12/18/2017 09:52 PM - bmbouter

- Subject changed from Repository Versions tracker to As a user, I have Repository Versions

#2 - 12/18/2017 11:21 PM - dkliban@redhat.com

- Description updated

#3 - 12/19/2017 03:20 AM - dkliban@redhat.com

- Description updated

#4 - 12/19/2017 01:00 PM - dkliban@redhat.com

- Description updated

#5 - 12/19/2017 01:48 PM - dkliban@redhat.com

- Description updated

#6 - 01/03/2018 11:25 PM - bmbouter

- Description updated

Rewriting the description with details.

#7 - 01/03/2018 11:28 PM - bmbouter

- Description updated

Adding in the POST to create a RepositoryVersion in the description.

#8 - 01/10/2018 08:28 PM - dkliban@redhat.com

- Description updated

#9 - 01/10/2018 08:33 PM - dkliban@redhat.com

- Description updated

#10 - 01/18/2018 02:39 PM - amacdona@redhat.com

- Description updated

#11 - 01/23/2018 08:21 PM - amacdona@redhat.com

- Description updated

#12 - 01/23/2018 09:13 PM - amacdona@redhat.com

- Description updated

#13 - 01/23/2018 09:58 PM - amacdona@redhat.com

- Description updated

#14 - 04/04/2018 08:51 PM - amacdona@redhat.com

- Status changed from NEW to MODIFIED

- Assignee set to amacdona@redhat.com

This was a parent task. The planning work is complete, the docs stories are groomed and ready to be added to the sprint.

#15 - 04/19/2018 05:12 PM - dkliban@redhat.com

- Sprint/Milestone set to 3.0.0

#16 - 04/26/2019 10:37 PM - bmbouter

- Tags deleted (Pulp 3, Pulp 3 MVP)

#17 - 12/13/2019 06:25 PM - bmbouter

- Status changed from MODIFIED to CLOSED - CURRENTRELEASE