

Arvados - Feature #13109

Support collection versions

02/21/2018 06:40 PM - Tom Morris

| | | | |
|------------------------|-------------------|------------------------|-----------|
| Status: | Resolved | Start date: | |
| Priority: | Normal | Due date: | |
| Assigned To: | | % Done: | 0% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | 2018-11-28 Sprint | | |

Description

See [Collection version history](#)

For some types of collections, particularly things like reference data, it is desirable to keep old versions around if they are updated.

User-facing features

- A collection has a current version number, so that the pair (uuid, version_nr) is be enough as a reference to a specific version of a particular collection.
- Whenever a collection get its manifest_text, description, properties or name fields updated, a new version is created (a 'snapshot' of the collection-to-be-changed record is created pointing to the updated, most current version.)
- The user can request a collection via an API call that includes past versions.
- The user can search on collections including past versions.
- Whenever a collection changes owner, uuid, storage classes, replication levels and trashed status, its past versions follow it.
- In order to modify a past version, the user needs to copy it into a new collection.

On workbench

- A new 'History' tab show the currently viewed collection position on a list of versions.
- On the 'History' tab, versions have the possibility to be copied to an entirely new collection and also to be copied as the current version (revert feature) on the history
- The collection's main pane show an indication if its the current version or an old one.

System wide configurations

- Flag to enable version history retention (OFF by default)

Implementation details

- All past versions go on the same collections table (so it's easier to do paging)
- New column current_version_uuid to hold the current version's UUID.
- New column version_number to hold a consecutive integer, starting at 1 for new collections.
- The following fields are synced with their past versions counterparts: replication_*, storage_classes_*, trash_at/delete_at/is_trashed, owner_uuid, uuid (update current_uuid to retain database consistency)
- Old versions with the same name shouldn't conflict with each other or other collections.

Related issues:

| | | |
|--|-----------------|-------------------|
| Related to Arvados - Feature #13561: [API] Store, and add APIs to retrieve, p... | Resolved | 10/04/2018 |
| Related to Arvados - Story #14086: [keep-web] Serve previous collection versions | Resolved | 10/15/2018 |
| Related to Arvados - Story #14299: [keep-balance] Ensure blocks referenced by... | Resolved | 10/23/2018 |

History

#1 - 02/21/2018 09:12 PM - Lucas Di Pentima

- Collections could have a flag that activates a copy-on-write behavior, maybe with a configurable default value.
- The collection update operation could also have a flag to override the collection's copy-on-write setting. This would allow FUSE driver to have some control over when to checkpoint modifications.
- Collections also would need a field to point to their ancestor collection, being filled with its own UUID or NULL (the latter would allow an easier database migration) when it's created from scratch.
- When copy-on-write is active and the collection is updated, before doing so it should be copied and saved as a new collection, and its UUID be used to update the ancestor field of the collection being updated. This way we'll have the UUID as a reference to the most recent version and the

PDH as reference to the exact version (like git, commit hashes versus branches).

- We may also need a way to distinguish between current versus old versions, so that they can be filtered out from project listings.
 - In the case of the user needing to restore an old version to some project, would the standard “copy to project” action be enough?
- For simplicity’s sake, we could start with a linear history:
 - Make old version collections immutable to avoid creating alternate version branches
 - Add a descendant_uuid field that’s filled up on creation for easy history forward-navigation. This would also allow to distinguish old versions (descendant_uuid != NULL) supporting the above feature.
 - Moving a collection to a new project would also include its ancestors
 - When trashing a collection, its ancestors should be trashed as well
- Things to consider:
 - Should search optionally include the old versions?
 - What should happen with attributes like name (collisions), description, properties, replication_desired?
 - Should the future storage_classes_desired attribute be treated the same way as trash_at and owner_uuid?
 - On the UI side (maybe this is for a separate story):
 - Should a collection view show the entire history or its immediate ancestor/descendant, taking into account the performance implications?
 - If the user is viewing an old version, some actions should be visibly disabled (eg: Move to project)

#2 - 04/04/2018 03:58 PM - Lucas Di Pentima

- Description updated

#3 - 05/16/2018 03:17 PM - Tom Clegg

- Related to Story #13494: Browse previous versions of a collection added

#4 - 06/04/2018 04:40 PM - Tom Clegg

- Related to Feature #13561: [API] Store, and add APIs to retrieve, previous versions of collection objects added

#5 - 07/11/2018 02:48 PM - Tom Clegg

- Description updated

#7 - 08/08/2018 03:58 PM - Lucas Di Pentima

- Description updated

#8 - 08/08/2018 04:23 PM - Lucas Di Pentima

- Description updated

#9 - 08/08/2018 04:30 PM - Lucas Di Pentima

- Description updated

#10 - 08/21/2018 05:42 PM - Tom Clegg

- Related to Story #14086: [keep-web] Serve previous collection versions added

#11 - 10/03/2018 04:09 PM - Tom Clegg

- Related to Story #14299: [keep-balance] Ensure blocks referenced by old collection versions are safe from garbage collection added

#12 - 10/31/2018 03:28 PM - Tom Morris

- Related to deleted (Story #13494: Browse previous versions of a collection)

#13 - 10/31/2018 03:31 PM - Tom Morris

- Status changed from New to Resolved

#14 - 12/12/2018 05:18 PM - Tom Morris

- Target version changed from To Be Groomed to 2018-11-28 Sprint

- Release set to 14