Arvados - Story #9278

[Crunch2] Document/fix handling of collections with non-nil expires_at field

05/24/2016 06:57 PM - Tom Clegg

Status:	In Progress	**Start date:**	06/01/2016
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Arvados Future Sprints		

Description

Draft desired behavior: Expiring collections

Current behavior:

- When deleting collections from the project view, Workbench sets expires_at=now().
- API server *does not* return expired collections in list or get responses. default_scope where("expires_at IS NULL or expires_at > CURRENT_TIMESTAMP")
- API server *does* return expiring collections in list responses.

Subtasks:

Task # 9298: Update arv-mount to desired behavior	**Closed**
Task # 9297: Update Workbench to desired behavior	**Closed**
Task # 9293: Review docs/spec [[Expiring collections]]	**Resolved**
Task # 9294: Document desired behavior and interpretation	**Resolved**
Task # 9296: Update API to desired behavior	**Resolved**
Task # 9302: Review 9278-expiring-collections	**Resolved**

Related issues:

Related to Arvados - Feature #9364: [keep-balance] "Expedited delete" tool: p...	**New**	
Related to Arvados - Story #9277: [Crunch2] System-owned container outputs sh...	**Resolved**	**02/16/2017**
Related to Arvados - Feature #3900: [Workbench] Trash button on collection us...	**Resolved**	**02/10/2017**
Related to Arvados - Story #9582: [Workbench] Don't display trashed collectio...	**Resolved**	**07/12/2016**
Related to Arvados - Story #9584: [FUSE] Don't display expiring collections i...	**New**	**07/12/2016**
Related to Arvados - Story #9587: [Workbench] Interface to list and untrash t...	**Resolved**	**07/12/2016**
Related to Arvados - Story #9589: [Workbench] Update collection interface for...	**Closed**	**07/13/2016**
Related to Arvados - Story #9590: [FUSE] Trash directory to list, inspect, an...	**New**	**07/13/2016**
Related to Arvados - Story #9591: [FUSE] Undelete collections by moving them ...	**New**	**07/13/2016**
Related to Arvados - Story #9592: [FUSE] rmdir on CollectionDirectory sets ex...	**New**	**07/13/2016**

Associated revisions

Revision 427da800 - 06/09/2016 02:21 PM - Tom Clegg

Merge branch '9278-expiring-collections'

refs #9278

History

#1 - 05/25/2016 07:19 PM - Tom Clegg

- Assigned To set to Tom Clegg

- Target version set to 2016-06-08 sprint

- Story points set to 2.0

#2 - 06/01/2016 03:05 PM - Tom Clegg

Draft at Expiring collections

#3 - 06/01/2016 06:53 PM - Brett Smith

Tom Clegg wrote:

Draft at Expiring collections

Thoughts:

At this point, I'm convinced that sysadmins need some way to accelerate deletion of blocks. Say someone accidentally uploads something much larger than the cluster was specced to store: usually what people want in this case is to delete that newly-created collection and blocks uniquely associated with it. A rule as simple as "Admins can set expires_at to an arbitrary time" would be sufficient to make this possible. Can the proposal grow to accommodate something like that? (I realize we need a way to solve this on the Keepstore end, too, but for now I think I'd be happy as long as changes on the API end don't make anything harder.)

More generally, I do wonder what ways you intend for it to be possible for clients to set or change expires_at, besides the existing delete method.

About the idea of creating collections to hold references to unreferenced blocks: I'm concerned about the ops impact of creating potentially many collections like this. It's also a little redundant: we already have the data we need in the Logs table. How would you feel about extending keep-balance to read those logs and use it as a new data source? The rule would be something like "Read all Logs for collection updates from the past [TTL duration]. Any block referenced in a manifest_text in any of those records is not eligible for deletion." If I've thought this through correctly, that would seem to eliminate any need for separate tracking of collection manifest changes. It would also correctly handle updates that change replication_desired from >0 to 0.

"In any case, an application that undeletes collections must be prepared to encounter name conflicts." - Will clients be able to just set ensure_unique_name=True to DTRT? If not, can we make that possible?

#4 - 06/01/2016 08:31 PM - Tom Clegg

- Description updated

#5 - 06/01/2016 09:56 PM - Peter Amstutz

Design comments:

expires_at	significance	get (pdh)	get (uuid)	appears in default list	can appear in list when filtering by expires_at
>now	expiring	yes(*)	yes(*)	no(**)	yes

(**) Change to "yes" after updating clients (arv-mount and Workbench) to behave appropriately, i.e., either use an expires_at filter when > requesting collection lists, or skip over them in default views.

On the principal of least surprise I would suggest settling on the behavior listed in the table and not plan on changing it as mentioned in (**). API clients that would need to care are not limited to Workbench and arv-mount but also crunch scripts written by users. API clients that care about expiring collections (workbench, block manager) can set the expires_at filter accordingly.

1. When expiring a collection, stash the original name somewhere and change its name to something unique (e.g., incorporating uuid and timestamp).

This is a little wonky but I'd be fine with that; I believe we have a "properties" hash on collections already. Partial indexes sound a bit tricky to set up.

Questions/clarifications:

What happens if a collection is deleted twice? Does expires_at get updated on the second delete?

Do you undelete a collection by setting expires_at to null? What happens if the user tries to do that on a collection that is past its expiration date?

What happens if a user is trying to delete a Project and there are expiring/expired collections?

Can expiring collections have their manifest text or other fields be updated? (probably not...)

#6 - 06/02/2016 06:38 PM - Tom Clegg

Peter Amstutz wrote:

Design comments:

expires_at	significance	get (pdh)	get (uuid)	appears in default list	can appear in list when filtering by expires_at
>now	expiring	yes(*)	yes(*)	no(**)	yes

> (**) Change to "yes" after updating clients (arv-mount and Workbench) to behave appropriately, i.e., either use an expires_at filter when requesting collection lists, or skip over them in default views.

> On the principal of least surprise I would suggest settling on the behavior listed in the table and not plan on changing it as mentioned in (**). API clients that would need to care are not limited to Workbench and arv-mount but also crunch scripts written by users. API clients that care about expiring collections (workbench, block manager) can set the expires_at filter accordingly.

Interesting. I like the "least surprise" principle, but I see it the other way around: the least surprising behavior is for "list without filters" to return all of the items, like it does elsewhere.

Adding an "expires_at is null" filter to an API request seems really easy, compared to combining the results of two (multi-page) queries using different filters.

Another option is to do what we do with keep services: one API for "list all", and one API for "list the ones I think you want".

The current behavior here is "yes", fwiw. It hasn't caused any confusion because it never comes up: nobody ever sets expires_at to a time in the future.

> 1. When expiring a collection, stash the original name somewhere and change its name to something unique (e.g., incorporating uuid and timestamp).

> This is a little wonky but I'd be fine with that; I believe we have a "properties" hash on collections already. Partial indexes sound a bit tricky to set up.

We already rely on partial indexes, so I don't think the setup trickery should be a factor.

> What happens if a collection is deleted twice? Does expires_at get updated on the second delete?

Earlier of {default expiry time} and {existing expiry time}, I think. (updated wiki)

"The given expires_at cannot be sooner than the existing expires_at and sooner than now+blobSignatureTTL."

> Do you undelete a collection by setting expires_at to null?

Yes. (updated wiki)

> What happens if the user tries to do that on a collection that is past its expiration date?

404. (updated wiki)

> What happens if a user is trying to delete a Project and there are expiring/expired collections?

Expired: they're invisible, so deleting a project should work exactly the same way it would without them.

Expiring: Not sure about this one. Dumping them in the parent project is one possibility.

> Can expiring collections have their manifest text or other fields be updated? (probably not...)

I don't think there's any reason to disallow this. One of the use cases is "scratch space". A rule "can't update expiring collection" would merely force clients to jump through undelete-modify-delete hoops, introducing the possibility of crashing in the middle and wasting storage space indefinitely.

#7 - 06/03/2016 07:47 PM - Tom Clegg

- Status changed from New to In Progress

#8 - 06/07/2016 01:29 PM - Tom Clegg

Changes needed:

- Clients: Use 'expires_at is null' filter in arv-mount and workbench (separate story: "show trash" feature for both)
- Clients: set expires_at=(now+permissionTTL) (or now+defaultTrashLifetime) instead of deleting collections outright
- API: Use shorter permission TTL in API response for expiring collection
- API: enforce expires_at >= now during update
- (#9363) handle deletion races: keep-balance needs to look in the logs table
- (#9364) expedited delete feature

#9 - 06/08/2016 02:04 PM - Tom Clegg

9278-expiring-collections @ 5c11190 includes

- 5c11190 9278: Ensure locator signatures expire no later than expires_at.
- 1a0738a 9278: Expose expires_at in API response.
- 67d893f 9278: Set expires_at=now if a client sets it to a time in the past.

#10 - 06/08/2016 07:09 PM - Tom Clegg

- Target version changed from 2016-06-08 sprint to 2016-06-22 sprint

#11 - 06/08/2016 07:10 PM - Tom Clegg

- Story points changed from 2.0 to 1.0

#12 - 06/09/2016 02:20 PM - Peter Amstutz

9278-expiring-collections @ 5c11190dda23801d0f7d177bf2c0a0ac5d899898 LGTM

#13 - 06/22/2016 08:03 PM - Tom Clegg

- Target version changed from 2016-06-22 sprint to Arvados Future Sprints

#14 - 07/12/2016 06:13 PM - Brett Smith

- Assigned To changed from Tom Clegg to Brett Smith

I will split off the subtasks into their own stories and make sure the desired behavior is specified in them directly.

#15 - 07/13/2016 07:37 PM - Brett Smith

Tom Clegg wrote:

- Clients: Use 'expires_at is null' filter in arv-mount and workbench

Workbench #9582, FUSE #9584.

(separate story: "show trash" feature for both)

Workbench #9587 (see also #9589), FUSE #9590 and #9591.

- Clients: set expires_at=(now+permissionTTL) (or now+defaultTrashLifetime) instead of deleting collections outright

Workbench #3900, FUSE #9592.

#16 - 10/12/2016 07:31 PM - Tom Morris

- Assigned To deleted (Brett Smith)

#17 - 04/25/2018 03:12 PM - Tom Morris

- Release deleted (11)