Arvados - Feature #12626

[API] Merge user accounts (redirect=true case)

11/22/2017 04:21 PM - Peter Amstutz

Status:	Resolved	**Start date:**	05/03/2018
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2018-05-09 Sprint		

Description

New API endpoint: POST /arvados/v1/users/merge

- Authorization header has valid API token for the "old" account
- new_user_token (post form param in request body) has valid API token for the "new" account
- new_owner_uuid (post form param in request body) has either new user's UUID, or a group UUID writable by the new user
- redirect_to_new_user=true (optional)

Security checks

- Current token ("old account") has scopes=["all"]
- new_user_token ("new account") has scopes=["all"]
- API logs show the UUID of the corresponding api_client_auth record instead of merge_into_token

Actions

- Move all records (groups, links, collections, jobs, pipelines, container requests, etc) owned by the old user into new_owner_uuid (this is typically a new empty project or a new user who doesn't own anything, so name conflicts would be a surprise/error)
- Update links set tail_uuid=new_user_uuid where tail_uuid=old_user_uuid

Additional actions if redirect_to_new_user=true

- Set old user's redirect_to_user_uuid field to the new user's UUID
- Move old user's SSH keys to the new user
- Ensure API tokens associated with old user will give access to the new account.
- Update links with head_uuid = old user to point to new user

...if redirect_to_new_user=false

- Leave old user's redirect_to_user_uuid field alone
- Delete old user's SSH keys
- Leave old user's API tokens alone
- Leave links with head_uuid = old user alone.

This is all done in a transaction: if anything fails, the entire operation is cancelled.

Implementation

- New column (users.redirect_to_user_uuid) is needed.
- #12995 and #12703 are blocked only by the redirect_to_new_user=true case.

Subtasks:

Task # 13397: Review 12626-merge-accounts	**Resolved**

Related issues:

Related to Arvados - Feature #4637: [SSO] Use "authentications" table and sup...	**Rejected**	**11/06/2017**
Related to Arvados - Story #12702: Migrate user accounts	**Resolved**	**01/05/2018**
Related to Arvados - Bug #13368: [API] Add "authorizations" table	**Closed**	
Blocks Arvados - Story #12703: [Workbench] Self serve account merge	**Resolved**	
Blocks Arvados - Story #12995: [Workbench] Allow user to add a new Google acc...	**Resolved**	**05/17/2018**

Associated revisions

Revision 945621a7 - 05/08/2018 01:30 PM - Tom Clegg

Merge branch '12626-merge-accounts'

refs #12626

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tclegg@veritasgenetics.com>

History

#1 - 11/22/2017 04:57 PM - Peter Amstutz

- Description updated

#2 - 11/29/2017 06:24 PM - Tom Morris

- Target version set to To Be Groomed

#3 - 11/29/2017 06:25 PM - Tom Morris

- Related to Feature #4637: [SSO] Use "authentications" table and support account linking added

#5 - 11/29/2017 07:48 PM - Peter Amstutz

- Related to Story #12702: Migrate user accounts added

#6 - 11/29/2017 07:56 PM - Peter Amstutz

- Description updated

#7 - 11/29/2017 07:57 PM - Peter Amstutz

- Description updated

#8 - 11/29/2017 07:58 PM - Peter Amstutz

- Blocks Story #12703: [Workbench] Self serve account merge added

#10 - 01/10/2018 08:30 PM - Tom Clegg

related: https://docs.google.com/document/d/1LdKscxuPbBnK4qX2ZHUQpSRhuDg8IEHJh_W2y8Y0-WY

#11 - 01/24/2018 07:28 PM - Tom Morris

- Target version changed from To Be Groomed to Arvados Future Sprints

#12 - 03/21/2018 06:57 PM - Tom Morris

- Related to Story #12995: [Workbench] Allow user to add a new Google account to their Arvados account added

#13 - 03/21/2018 09:01 PM - Tom Clegg

- Description updated

#14 - 03/21/2018 09:03 PM - Tom Morris

- Related to deleted (Story #12995: [Workbench] Allow user to add a new Google account to their Arvados account)

#15 - 03/21/2018 09:04 PM - Tom Morris

- Blocks Story #12995: [Workbench] Allow user to add a new Google account to their Arvados account added

#16 - 04/18/2018 03:30 PM - Tom Clegg

- Subject changed from Merge user accounts to [API] Merge user accounts

- Description updated

#17 - 04/18/2018 07:30 PM - Tom Clegg

- Related to Bug #13368: [API] Add "authorizations" table added

#18 - 04/18/2018 07:58 PM - Tom Clegg

- Description updated

#19 - 04/18/2018 07:59 PM - Tom Morris

- Story points set to 3.0

#20 - 04/25/2018 03:36 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2018-05-09 Sprint

#21 - 04/25/2018 03:53 PM - Tom Morris

- Assigned To set to Tom Clegg

#22 - 05/01/2018 03:39 PM - Tom Clegg

- Status changed from New to In Progress

#23 - 05/02/2018 07:11 PM - Tom Clegg

- Description updated

#24 - 05/03/2018 02:08 PM - Tom Clegg

12626-merge-accounts @ 26538afdf1c8fdad14208d08a19bafb41e42044c

#25 - 05/04/2018 04:02 PM - Peter Amstutz

- Distinguishing between "new user" and "old user" is confusing enough without UsersController#merge using the term "dst_user". User#merge also has a comment "# Merge this user's owned items into dst_user." but the term dst_user isn't used anywhere in that method. Please use consistent terminology.
- Please add comments to UsersController#merge explaining the respective roles of "current user" and "new user".
- I'm confused what this is checking for. I think it is checking that the new user has write access to the new owner (which I guess could be either the user itself, or a group). Also seems to be missing a default (or a check) when 'new_owner_uuid' is not supplied? Needs a comment.

```
    if !dst_user.can?(write: params[:new_owner_uuid])
      return send_error("new_owner_uuid is not writable", status: 403)
    end
```

- This seems to assume new_owner_uuid is a user, but it could also be a group?

```
      if User.where('uuid in (?) and redirect_to_user_uuid is not null',
                    [new_owner_uuid, redirect_to_user_uuid]).any?
        raise "cannot merge to/from an already merged user"
      end
```

- Purely from a formatting standpoint, this is confusing, it looks like the line starting with "[" is part of the previous statement (could use a blank line between statements).

```
      ApiClientAuthorization.
        where(user_id: id).
        update_all(user_id: new_user.id)
      [
        [AuthorizedKey, :owner_uuid],
```

- Why is explicit update of "AuthorizedKey.owner_uuid" &c necessary, won't they be updated by "change_all_uuid_refs"? → On further inspection I see this is because the first part updates to the user, the second one updates to the new owner. But the second one actually updates anything ending in _uuid, where we probably only want to update owner_uuid ?

#26 - 05/04/2018 07:13 PM - Tom Clegg

Peter Amstutz wrote:

> - Distinguishing between "new user" and "old user" is confusing enough without UsersController#merge using the term "dst_user". User#merge also has a comment "# Merge this user's owned items into dst_user." but the term dst_user isn't used anywhere in that method. Please use consistent terminology.

Updated comment and controller method.

> - Please add comments to UsersController#merge explaining the respective roles of "current user" and "new user".

(included in updated comment)

- I'm confused what this is checking for. I think it is checking that the new user has write access to the new owner (which I guess could be either the user itself, or a group). Also seems to be missing a default (or a check) when 'new_owner_uuid' is not supplied? Needs a comment.

Updated the error message instead of adding a comment, since I figure the purpose of the check should be obvious from the error message if the error message is any good.

```
    if !new_user.can?(write: params[:new_owner_uuid])
      return send_error("
cannot move objects into supplied new_owner_uuid: new user does not have write permission", status: 403)
    end
```

Added tests for empty/missing new_owner_uuid.

Added a _merge_requires_parameters method for the discovery doc.

- This seems to assume new_owner_uuid is a user, but it could also be a group?

[...]

Oops, that should have been self.uuid, not new_owner_uuid. Fixed, and split old/new user checks to make the error messages more specific.

- Purely from a formatting standpoint, this is confusing, it looks like the line starting with "[" is part of the previous statement (could use a blank line between statements).

[...]

Added blank line.

- Why is explicit update of "AuthorizedKey.owner_uuid" &c necessary, won't they be updated by "change_all_uuid_refs"? → On further inspection I see this is because the first part updates to the user, the second one updates to the new owner. But the second one actually updates anything ending in _uuid, where we probably only want to update owner_uuid ?

Yes... changing modified_by_user_uuid etc. shouldn't be necessary. Fixed so it just touches owner_uuid, and added comments.

12626-merge-accounts @ 4cbac38547d8047e5e23cb4945b25aaa31e3eb06

#27 - 05/07/2018 06:12 PM - Peter Amstutz

So I tried using the merge API using arv and the behavior is still rather confusing:

- The current "merge" API currently means roughly "give away all my stuff".

For the use case we have, we have an old login, and we want to associate a new login with the old login. The way I do this is log in to the new account and give away my stuff to the old account, using new_owner=old_account.

Once I've given away my stuff, the new token I used to give it a way indicates I'm now a new (old) user. The old (new) user becomes hidden.

This is a little bit confusing. Maybe "source" and "target" ?

- It needs to trigger a permission graph update after reassigning ownership.
- When you log in with the old account, the user record gets the name and email address of the old account. However, if you log in with the new (source) account, the user record gets the name and email address from the new account. This causes flapping in the user record if the user logs in both ways.

#28 - 05/07/2018 07:05 PM - Tom Clegg

The current API makes more sense if you think "merge" means "merge me [into account X]". The reverse API would make more sense if you think "merge" means "merge [account X] into me". I think with this sort of thing we just have to pick one way and expect client devs to read the label if they care which way is up.

"Old" and "new" refer to the fact that one of the accounts is going away (the "old" one) and you'll be using the other one (the "new" one) instead in the future.

Is this what you have in mind for "target"?

New API endpoint: POST /arvados/v1/users/merge

- Authorization header has valid API token for the current ("old") account
- target_user_token (post form param in request body) has valid API token for the "target" account

- target_owner_uuid (post form param in request body) has either target user's UUID, or a group UUID writable by the target user
- redirect_to_target_user=true (optional)

(The term "old" isn't even part of the API so I suggest we not worry too much about it.)

#29 - 05/07/2018 07:17 PM - Tom Clegg

12626-merge-accounts @ 931f77f9bff46dbba8adb0517720eb3c60b83bb3

- permission graph update
- update email on auth user, not target user, during auth with a merged/redirected account

#30 - 05/07/2018 08:03 PM - Peter Amstutz

Tom Clegg wrote:

> The current API makes more sense if you think "merge" means "merge me [into account X]". The reverse API would make more sense if you think "merge" means "merge [account X] into me". I think with this sort of thing we just have to pick one way and expect client devs to read the label if they care which way is up.
>
> "Old" and "new" refer to the fact that one of the accounts is going away (the "old" one) and you'll be using the other one (the "new" one) instead in the future.
>
> Is this what you have in mind for "target"?
>
> New API endpoint: POST /arvados/v1/users/merge
>
> - Authorization header has valid API token for the current ("old") account
> - target_user_token (post form param in request body) has valid API token for the "target" account
> - target_owner_uuid (post form param in request body) has either target user's UUID, or a group UUID writable by the target user
> - redirect_to_target_user=true (optional)
>
> (The term "old" isn't even part of the API so I suggest we not worry too much about it.)

Yea, forget I said anything. It's confusing but I don't want to derail on this.

#31 - 05/08/2018 01:21 PM - Peter Amstutz

Tom Clegg wrote:

> 12626-merge-accounts @ 931f77f9bff46dbba8adb0517720eb3c60b83bb3
>
> - permission graph update
> - update email on auth user, not target user, during auth with a merged/redirected account

I did another manual test and it is working the way I expected now.

LGTM, please merge.

#32 - 05/09/2018 01:17 PM - Tom Clegg

- Subject changed from [API] Merge user accounts to [API] Merge user accounts (redirect=true case)

- Status changed from In Progress to Resolved

#33 - 07/23/2018 06:41 PM - Tom Morris

- Release set to 13