Arvados - Feature #15061

Redirect users to log in with correct federated identity

04/02/2019 08:47 PM - Peter Amstutz

Status:	Resolved	**Start date:**	04/18/2019
Priority:	Normal	**Due date:**	
Assigned To:	Peter Amstutz	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2019-06-05 Sprint		

Description

New design, based on discussion Apr 17

Existing user:

1. When a user logs into a home cluster, make ajax calls to known federated cluster login endpoints to say "this browser prefers cluster X as home" which returns a cookie.
2. User arrives at a federated cluster. The login button takes user to login endpoint on API server. User can also choose a specific home cluster for log in.
3. Request to login endpoint includes cookie saying user prefers cluster X, which can be overridden with an explicit query parameter that indicates a home cluster
4. API server redirects login to proper home cluster

Migrating to remote account (because you have an existing account or created one by accident)

1. User logs in to local account
2. User selects "migrate account" and selects home cluster X
3. Current token is saved in local session storage and user is redirected to log into cluster X
4. User is redirected back to cluster with salted token from cluster X
5. Everything owned by local user is reassigned to remote user and local user is marked "redirect_to_user_uuid" to the remote
6. User now uses token as remote user

Logging into a redirected account, no cookies or other hints telling us which cluster to use:

1. User logs in to local account
2. After log in, we realize redirected user is not local
3. Display a page that says "this has been migrated to a remote account, must log in at home cluster"
4. Redirect to home cluster
5. User logs in a second time (Existing user flow)

Scripted user migration

1. Admin generates list of email address and/or usernames assigned to each home cluster
2. Get list of users on each cluster
3. If there are user records with the email address or username that doesn't match the assigned home cluster, perform account merge
4. Need to tweak "merge" endpoint for admin variant which accepts "old_user_uuid" and "new_user_uuid" instead of using current token / "new_user_token"

Subtasks:

Task # 15089: Detailed design	**Resolved**
Task # 15140: API server updates	**Resolved**
Task # 15141: Workbench2 updates	**Resolved**
Task # 15142: Review 15061-fed-login	**Resolved**
Task # 15208: Migration script	**Resolved**
Task # 15219: Review 15061-fed-migrate	**Resolved**
Task # 15221: Review workbench2 updates	**Resolved**

Related issues:

Related to Arvados - Story #15088: [Workbench2] Replicate Workbench1 merge ac...	**Resolved**	**05/02/2019**
Related to Arvados - Feature #15064: [Workbench2] Use long-lived cookies to i...	**Resolved**	**05/14/2019**

Related to Arvados - Feature #15531: [SDK] Migrate federation to central Logi...	Resolved	09/23/2019

Associated revisions

Revision 7881bce4 - 05/08/2019 07:45 PM - Peter Amstutz

arvbox crunch-run and certificate fixes

arvbox crunch-run change to -container-enable-networking=default

Previously was "always" which causes CWL tests that checked that
networking was disabled to fail.

arvbox root-cert creates file with .crt instead of .pem, because
that's the file extension update-ca-certificates looks for.

Add cluster id and timestamp to arvbox test certificate common name to
prevents collisions on the certificate subject.

Arvbox sets trusted api_client for workbench2.

refs #15028 refs #15061

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision 9eca0b4f - 05/09/2019 08:57 PM - Peter Amstutz

Merge branch '15061-fed-login' refs #15061

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision ef5ce10f - 05/15/2019 08:09 PM - Peter Amstutz

Merge branch '15061-fed-migrate' refs #15061

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision d8622da3 - 05/17/2019 02:19 PM - Peter Amstutz

Add user email and cancel button to federated user redirect page

refs #15061

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision 2b71ad23 - 05/21/2019 02:16 PM - Peter Amstutz

Make SSO sessions expire after 1 minute.

Whenever the user arrives at the SSO server, they should get a login
screen. Avoid "sticky" sessions where the user is automatically
logged in with the existing session but needed to log in as another
user (specifically account merging/linking.)

refs #15061

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision a40c8289 - 05/21/2019 03:49 PM - Peter Amstutz

Merge branch '15061-session-timeout' refs #15061

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

History

#2 - 04/02/2019 09:50 PM - Peter Amstutz

- Description updated

#3 - 04/02/2019 09:56 PM - Peter Amstutz

- Description updated

#4 - 04/03/2019 02:13 PM - Peter Amstutz

- Description updated

#5 - 04/03/2019 02:15 PM - Peter Amstutz

- Description updated

#6 - 04/03/2019 02:17 PM - Peter Amstutz

- Description updated

- Story points set to 4.0

#7 - 04/03/2019 02:17 PM - Peter Amstutz

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

#8 - 04/03/2019 04:56 PM - Tom Clegg

- Related to Feature #15064: [Workbench2] Use long-lived cookies to improve login chooser defaults added

#9 - 04/10/2019 03:29 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2019-04-24 Sprint

#10 - 04/10/2019 03:35 PM - Tom Morris

- Related to Story #15088: [Workbench2] Replicate Workbench1 merge account feature added

#11 - 04/10/2019 03:41 PM - Tom Morris

- Description updated

#12 - 04/10/2019 03:52 PM - Tom Morris

- Description updated

- Target version changed from 2019-04-24 Sprint to To Be Groomed

#13 - 04/10/2019 03:57 PM - Tom Morris

- Target version changed from To Be Groomed to 2019-04-24 Sprint

#14 - 04/16/2019 08:40 PM - Peter Amstutz

1. When a user logs in / creates an account on cluster A (user_sessions_controller#create) send a simple federated request (like /arvados/v1/user/current) with a salted token to all clusters in the federation on behalf of the user. This will cause the remote clusters to cache a record of the remote user.
    1. Slightly awkward because we'd like controller to do this, but controller isn't hooked into the login process
    2. Controller sees the redirect from SSO back to API with the oauth2 one-off joshid auth token
    3. Then it sees the redirect response (with the API token)
    4. So controller could recognize the redirect response, intercept the token and spin off a thread to contact the other clusters
2. When a user logs in / creates an account on cluster B and the account is inactive, check if there are any cluster A user records with the same email address.
    1. The options here seem to be to either make a special case in the security model ("permitted to see user accounts with same email address as the current user account") or a special case dedicated API call that bypasses the security model "/arvados/v1/users/same_email_as_me"
3. Using whichever method is decided on, Workbench gets the user list of same email addresses (or other "probably same user" criteria) and presents the user with options to migrate accounts
    1. Sees that there is a pre-populated record from cluster A
    2. Prompt the user to migrate the (newly created) cluster B user to the cluster A user instead.
    3. This should come up automatically when the user is inactive, otherwise it is accessible through a menu item
    4. User is given a button to push that says "I meant to log in as cluster A user"
    5. This stores the token in session storage and sends the user to perform the login process again at cluster A
4. User logs at cluster A and is redirected back with a token salted for cluster B.
    1. Return to the "merge/migrate account" page with the new token, and the old token in session storage
    2. Give the user a button that says "finish logging in" which actually uses the merge API to migrate the account (setting redirect_to_user_uuid to the cluster A user) the user finishes logging in.
    3. Set a parameter in local storage that indicates the user should log in using cluster A, this should make the UI select cluster A by default in the future.
    4. The user ends up at workbench B with a salted token.

Variant: user chooses to attempt to log as a local user to cluster B **again** (maybe they are on a different computer) but meant to log in as federated user

1. User is authorized as existing user on B, which now has redirect_to_user_uuid set to the federated user on A

1. We know the (remote) user she intended to log in as
2. Issue a local token for that user? Authorizes as remote user at that cluster without communicating to A (home). Can't be used at federates, though (multi-site search won't work).

Questions to decide:

- What is the API for getting the list of users with same email address?
- What happens when you log into a cluster and the local user has redirect_to_user_uuid to a remote user?

#15 - 04/18/2019 03:05 PM - Peter Amstutz

- Description updated

#16 - 04/18/2019 03:33 PM - Tom Morris

- Target version changed from 2019-04-24 Sprint to To Be Groomed

- Story points deleted (4.0)

#17 - 04/18/2019 06:08 PM - Peter Amstutz

- Description updated

- Target version changed from To Be Groomed to 2019-04-24 Sprint

- Story points set to 4.0

#18 - 04/18/2019 06:32 PM - Tom Morris

- Target version changed from 2019-04-24 Sprint to Arvados Future Sprints

- Story points changed from 4.0 to 5.0

#19 - 04/24/2019 04:45 PM - Tom Morris

- Target version changed from Arvados Future Sprints to 2019-05-08 Sprint

#20 - 04/24/2019 05:08 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#21 - 04/24/2019 05:14 PM - Tom Morris

- Related to deleted (Feature #15064: [Workbench2] Use long-lived cookies to improve login chooser defaults)

#22 - 04/24/2019 05:16 PM - Tom Morris

- Related to Feature #15064: [Workbench2] Use long-lived cookies to improve login chooser defaults added

#23 - 04/25/2019 03:26 PM - Tom Morris

How do we handle the case where a user's home cluster is decommissioned? Do we need a migration utility to migrate accounts to a different cluster?

#24 - 04/25/2019 03:43 PM - Peter Amstutz

Tom Morris wrote:

> How do we handle the case where a user's home cluster is decommissioned? Do we need a migration utility to migrate accounts to a different cluster?

I don't know, but that's out of scope for this ticket. We should discuss it in a future engineering design meeting.

#25 - 05/06/2019 05:57 PM - Peter Amstutz

- Status changed from New to In Progress

#26 - 05/08/2019 01:30 PM - Peter Amstutz

Setting cookies (notes)

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS?redirectlocale=en-US&redirectslug=HTTP_access_control#Requests_with_credentials

1. API server publishes CORS header "Access-Control-Allow-Credentials: true" (this enables cross-domain cookies)
2. API server adds a "browserhome" endpoint
3. To set browser home, make ajax call to https://api/browserhome?set=xyzab, withCredentials = true, this responds with "Set-Cookie:

browserhome=xyzab" (should this be GET or POST?)
4. To get browser home, make ajax call to https://api/browserhome, withCredentials = true, this will read the Cookie and send it back

(I don't think we can use document.cookie because the wb2 document is served on a different domain from the API server.)

#27 - 05/08/2019 03:51 PM - Peter Amstutz

Need to be able to migrate a remote user to a local user. This means reassigning ownership, permissions etc of the remote user to the local user.

The "redirect_to_user_uuid" feature only applies during the login process. However, for remote users, there's no login process on the (local) cluster. The user just shows up with a remote token.

This seems like a possible use case for merge with "redirect_to_new_user=false". The proposed behavior is to reassign ownership from the remote user to the local user except for things like API tokens and ssh keys.

However remote user account will still exist, and could be logged into again, which would be confusing. We could deactivate the remote user (but they could be reactivated, depending on the configuration). We could set "redirect_to_user_uuid" on the remote user record, although we don't actually want to redirect anything, this would effectively blacklist the remote user (and the error messages could tell you which user you're supposed to log in as).

#28 - 05/08/2019 05:21 PM - Peter Amstutz

- Target version changed from 2019-05-08 Sprint to 2019-05-22 Sprint

#29 - 05/08/2019 05:21 PM - Peter Amstutz

- Target version changed from 2019-05-22 Sprint to 2019-05-08 Sprint

Peter Amstutz wrote:

> Need to be able to migrate a remote user to a local user. This means reassigning ownership, permissions etc of the remote user to the local user.
>
> The "redirect_to_user_uuid" feature only applies during the login process. However, for remote users, there's no login process on the (local) cluster. The user just shows up with a remote token.
>
> This seems like a possible use case for merge with "redirect_to_new_user=false". The proposed behavior is to reassign ownership from the remote user to the local user except for things like API tokens and ssh keys.
>
> However remote user account will still exist, and could be logged into again, which would be confusing. We could deactivate the remote user (but they could be reactivated, depending on the configuration). We could set "redirect_to_user_uuid" on the remote user record, although we don't actually want to redirect anything, this would effectively blacklist the remote user (and the error messages could tell you which user you're supposed to log in as).

Not doing this on this ticket

#30 - 05/08/2019 05:24 PM - Peter Amstutz

- Target version changed from 2019-05-08 Sprint to 2019-05-22 Sprint

#31 - 05/08/2019 05:36 PM - Peter Amstutz

15061-fed-login @ 6eec8fe2f0adb2bd0f53a51a37ca8ebbeaced44d

#32 - 05/08/2019 06:23 PM - Peter Amstutz

- File users.csv added

#33 - 05/09/2019 03:16 PM - Lucas Di Pentima

15061-fed-login LGTM.

The only observation is that the sass-rails gem used give deprecation warnings while I was doing the rails5 upgrade, and there is a sassc-rails gem that is a drop-in replacement. It seems that they made a release wrapping sassc-rails, see:

- https://github.com/rails/sass-rails/issues/420
- https://github.com/rails/sass-rails/pull/424/commits

So maybe we want to directly ask for sassc-rails instead?

#34 - 05/09/2019 03:33 PM - Peter Amstutz

Lucas Di Pentima wrote:

> 15061-fed-login LGTM.
>
> The only observation is that the sass-rails gem used give deprecation warnings while I was doing the rails5 upgrade, and there is a sassc-rails gem that is a drop-in replacement. It seems that they made a release wrapping sassc-rails, see:
>
> - https://github.com/rails/sass-rails/issues/420
> - https://github.com/rails/sass-rails/pull/424/commits
>
> So maybe we want to directly ask for sassc-rails instead?

It is working now, so I don't want to mess with it.

#35 - 05/09/2019 03:34 PM - Peter Amstutz

https://ci.curoverse.com/view/Developer/job/developer-run-tests/1230/

#36 - 05/09/2019 07:32 PM - Peter Amstutz

https://ci.curoverse.com/view/Developer/job/developer-run-tests/1231/

#37 - 05/09/2019 07:40 PM - Peter Amstutz

15061-fed-migrate @ 05bfa9a17e8d46d9a388fea130b7df33b7aa15c6

User migration script and documentation.

#38 - 05/10/2019 08:03 PM - Lucas Di Pentima

Still unable to run the script against my 2-arvbox federation because of SSL cert validation issues. Nevertheless, here're some comments:

- On documentation
  - Missing “to” in sentence: A federated user account is associated with a specific “home” cluster, and can be used access other clusters in the federation that trust the home cluster.
  - ‘at-sign‘ char in: “...for each user. In this example, person_b@example.com is assigned…"
- On arv-federation-migrate
  - I think some kind of arguments help messages would be very beneficial.
  - The link to the documentation on the parser description is not pointing to the correct page
  - Should it do an initial check to see that remote_hosts settings are equal on every cluster defined on tokens.csv and that all the remote_hosts are listed on the file? I think it would be useful to detect misconfigurations and omissions on the tokens file.
  - Is line 115 necessary to do the migration call? Or is it just to confirm that the salted token works?

#39 - 05/14/2019 06:15 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Still unable to run the script against my 2-arvbox federation because of SSL cert validation issues. Nevertheless, here're some comments:
>
> - On documentation
>   - Missing “to” in sentence: A federated user account is associated with a specific “home” cluster, and can be used access other clusters in the federation that trust the home cluster.

Fixed.

> - ‘at-sign‘ char in: “...for each user. In this example, person_b@example.com is assigned…"

Fixed.

> - On arv-federation-migrate
>   - I think some kind of arguments help messages would be very beneficial.

I added some help text. Let me know if you think it should say more.

> - The link to the documentation on the parser description is not pointing to the correct page

Fixed.

> - Should it do an initial check to see that remote_hosts settings are equal on every cluster defined on tokens.csv and that all the remote_hosts are listed on the file? I think it would be useful to detect misconfigurations and omissions on the tokens file.

Now it checks for well-connectedness among the clusters listed in the tokens file.

> - Is line 115 necessary to do the migration call? Or is it just to confirm that the salted token works?

It ensures that the remote user is listed so that it can be migrated to (otherwise it may fail if there is no user record).

#40 - 05/14/2019 06:15 PM - Peter Amstutz

15061-fed-migrate @ 81ca884e5ca26bec3bba79c94286e7923f3be82b

#41 - 05/14/2019 07:52 PM - Lucas Di Pentima

Still trying to manually test the script locally. It checked the --check argument against 4xphq & c97qk but didn't want to try the others arguments against those live clusters. In the meantime I have this comment:

- The --check argument ignores the case where a federated cluster is listed on remoteHosts, but is missing on the tokens file. Mentioning this just in case it isn’t what we want.

#42 - 05/14/2019 07:56 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Still trying to manually test the script locally. It checked the --check argument against 4xphq & c97qk but didn't want to try the others arguments against those live clusters. In the meantime I have this comment:
>
> - The --check argument ignores the case where a federated cluster is listed on remoteHosts, but is missing on the tokens file. Mentioning this just in case it isn’t what we want.

A cluster could be part of multiple federations, the goal here is to synchronize accounts among a single well-connected federation.

But it might make sense to generate a warning in that case, not a fatal error.

#43 - 05/14/2019 08:18 PM - Peter Amstutz

15061-fed-migrate @ ec96bfb7ddad8d27862ace0c382f9bd33679f96c

#44 - 05/14/2019 11:52 PM - Lucas Di Pentima

Finally got the test env working!

- Maybe it would be convenient to avoid accessing cached discovery document, it happened to me that one of the clusters’ discovery docs was cached with a misconfigured remoteHosts and the --check option was reporting something that wasn’t true.
- When generating the report, the admin users were included. From what I’m reading at https://dev.arvados.org/issues/12995#note-19, it may not be desirable to allow admin user migrations because admin access will be lost.

#45 - 05/15/2019 03:38 PM - Peter Amstutz

Lucas Di Pentima wrote:

> Finally got the test env working!
>
> - Maybe it would be convenient to avoid accessing cached discovery document, it happened to me that one of the clusters’ discovery docs was cached with a misconfigured remoteHosts and the --check option was reporting something that wasn’t true.

Ah, you mean the cached discovery document use by the arvados sdk (there is was a similar problem in workbench2). Will fix.

> - When generating the report, the admin users were included. From what I’m reading at https://dev.arvados.org/issues/12995#note-19, it may not be desirable to allow admin user migrations because admin access will be lost.

The report needs to include both users because when you fill in the "home cluster" column it actually goes down the list and finds the user uuid in the same document that corresponds to that email address and cluster in order to do the merge.

But it could check for the case where the 'old' account is admin and print an error unless both the target is also admin.

#46 - 05/15/2019 03:41 PM - Lucas Di Pentima

Additional comments:

- If the admin makes a mistake when writing the uuid prefix when assigning a home to some account, the error message is for example "No user listed to migrate x1wjp-tpzed-2ujf6av44sres3t to xfp4l”, when the correct prefix was xfp5l, this can lead to confusion when dealing with big lists.
- If re-running the migration with the same report (or some partial changes), there’s an api error that could be catched and reported without exiting

(maybe the admin forgot to call a new report and is making a correction to the last one): arvados.errors.ApiError: <HttpError 422 when requesting https://172.17.0.3:8000/arvados/v1/users/merge?new_owner_uuid=xfp5l-j7d0g-pk63hj4cf18bus7&old_user_uuid=xfp5l-tpzed-boyvp0lkap9uvr9&redirect_to_new_user=true&alt=json&new_user_uuid=x1wjp-tpzed-c8oreg11zbl14xt returned "User in old_user_uuid not found">

- It seems to be some kind of bug related to the order of which the tokens are listed on the tokens.csv file. Let’s suppose I have cluster1 & cluster2 on the federation, and a user@test.com account on both. I’m able to log in to both accounts on both clusters without issues. Let’s suppose on the tokens.csv file I have cluster1 at the 1st line, cluster2 at the 2nd. Then, I get the report with both user@test.com accounts listed. If I try to migrate cluster1’s account to cluster2, I get the following error: arvados.errors.ApiError: <HttpError 401 when requesting https://172.17.0.2:8000/arvados/v1/users/current?alt=json returned "Not logged in”> If I try the other way around, it works. Also, if I change the tokens order, I can do what I tried first.

#47 - 05/15/2019 07:01 PM - Peter Amstutz

I did a big code cleanup and more validity checks and error handling.

15061-fed-migrate @ 36a33fe214b13cbe9388df92fc7bf83c81f42d11

Lucas Di Pentima wrote:

> Additional comments:
>
> - If the admin makes a mistake when writing the uuid prefix when assigning a home to some account, the error message is for example "No user listed to migrate x1wjp-tpzed-2ujf6av44sres3t to xfp4l”, when the correct prefix was xfp5l, this can lead to confusion when dealing with big lists.

It should now offer a more useful error.

> - If re-running the migration with the same report (or some partial changes), there’s an api error that could be catched and reported without exiting (maybe the admin forgot to call a new report and is making a correction to the last one): arvados.errors.ApiError: <HttpError 422 when requesting https://172.17.0.3:8000/arvados/v1/users/merge?new_owner_uuid=xfp5l-j7d0g-pk63hj4cf18bus7&old_user_uuid=xfp5l-tpzed-boyvp0lkap9uvr9&redirect_to_new_user=true&alt=json&new_user_uuid=x1wjp-tpzed-c8oreg11zbl14xt returned "User in old_user_uuid not found">

This should be caught and produce a better error suggesting maybe the user was migrated already.

> - It seems to be some kind of bug related to the order of which the tokens are listed on the tokens.csv file. Let’s suppose I have cluster1 & cluster2 on the federation, and a user@test.com account on both. I’m able to log in to both accounts on both clusters without issues. Let’s suppose on the tokens.csv file I have cluster1 at the 1st line, cluster2 at the 2nd. Then, I get the report with both user@test.com accounts listed. If I try to migrate cluster1’s account to cluster2, I get the following error: arvados.errors.ApiError: <HttpError 401 when requesting https://172.17.0.2:8000/arvados/v1/users/current?alt=json returned "Not logged in”> If I try the other way around, it works. Also, if I change the tokens order, I can do what I tried first.

Yea I think what happened is that I was reusing the variable 'arv' and not reassigning it and so it would either work by accident or make an API call on the wrong cluster.

#48 - 05/15/2019 08:04 PM - Lucas Di Pentima

This LGTM, thanks!

#49 - 05/17/2019 01:43 PM - Peter Amstutz

Todo on api server redirect page (based on feedback / testing)

- email of the user being redirected to
- cancel button

#50 - 05/17/2019 02:53 PM - Peter Amstutz

Peter Amstutz wrote:

> Todo on api server redirect page (based on feedback / testing)
>
> - email of the user being redirected to
> - cancel button

Added this @ d8622da3183d2028b052e5c622635b96b6d4aa23

#51 - 05/21/2019 02:27 PM - Peter Amstutz

SSO server tweak. Fixes the following bug:

1. User goes to link account
2. Users logs in with a different account
3. User hits cancel
4. User starts link account again
5. User is automatically logged in with existing SSO session instead of

Solution in this branch is to give rails sessions a 1 minute timeout.

15061-session-timeout @ commit:2b71ad232495ece0403b5f717a561a38d6b8d0c6

#52 - 05/21/2019 02:40 PM - Lucas Di Pentima

15061-session-timeout LGTM. 1 minute expiration time seems like a good enough middle ground.

#53 - 05/22/2019 05:15 PM - Tom Morris

- Target version changed from 2019-05-22 Sprint to 2019-06-05 Sprint

#54 - 05/22/2019 09:50 PM - Peter Amstutz

- Status changed from In Progress to Resolved

#55 - 08/21/2019 05:09 PM - Tom Morris

- Related to Feature #15531: [SDK] Migrate federation to central LoginCluster added

Files

users.csv	9.36 KB	05/08/2019	Peter Amstutz