Arvados - Feature #12018

Synchronize group membership with external data source

07/21/2017 09:09 PM - Tom Morris

Status:	Resolved	**Start date:**	07/21/2017
Priority:	Normal	**Due date:**	
Assigned To:	Lucas Di Pentima	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2017-12-06 Sprint		

Description

As a user in a corporate environment, I want to be able to synchronize the users in my Arvados groups with my corporate directory service (ActiveDirectory, LDAP, etc).

This doesn't need to be instantaneous, but can instead by done either periodically on a scheduled based or on demand. A script-based solution is an acceptable answer.

Groups which get created by this mechanism get tagged so that they're known to be automatically created. Groups are not given any particular permissions when they are created.

Input is a two column CSV file with a column of Group name and one column of user IDs (either username or user email address) with a command flag which controls whether the user ID is username or email address. If a user is no longer included in the input file, they get removed from the group membership.

Workbench needs to be changed to not allow admins to modify group membership for synched.

Tool should report errors for any users who don't have matching user IDs. Groups which don't exist get created and their UUIDs get reported. If an untagged group exists and is also in the input file, a warning is issued.

Subtasks:

Task # 12264: Review 12018-sync-groups-tool	**Resolved**
Task # 12656: Review 12018-tool-docs	**Resolved**

Associated revisions

Revision cc6f86f1 - 11/02/2017 02:58 PM - Lucas Di Pentima

Merge branch '12018-sync-groups-tool'
Closes #12018

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

Revision 5b2ebfe3 - 11/29/2017 09:59 PM - Lucas Di Pentima

Merge branch '12018-tool-docs'
Refs #12018

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

Revision 0cbcf8cb - 12/05/2017 08:24 PM - Ward Vandewege

Rename the group sync tool to follow our standard naming scheme.

refs #12018

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <wvandewege@veritasgenetics.com>

Revision 60616882 - 12/05/2017 08:40 PM - Ward Vandewege

Do not hardcode the name of the executable in the sync-groups code.

refs #12018

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <wvandewege@veritasgenetics.com>

Revision 50b36937 - 12/05/2017 08:49 PM - Ward Vandewege

Merge branch '12018-rename-sync-groups'

refs #12018

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <wvandewege@veritasgenetics.com>

Revision ee8558dd - 12/05/2017 09:26 PM - Ward Vandewege

Update documentation to the new name of the sync-groups tool.

refs #12018

Arvados-DCO-1.1-Signed-off-by: Ward Vandewege <wvandewege@veritasgenetics.com>

History

#1 - 08/15/2017 07:11 PM - Tom Morris

- Description updated

- Target version changed from Arvados Future Sprints to 2017-08-30 Sprint

- Story points set to 2.0

#2 - 08/16/2017 07:32 PM - Tom Morris

- Target version changed from 2017-08-30 Sprint to 2017-09-13 Sprint

#3 - 08/30/2017 07:26 PM - Tom Morris

- Target version changed from 2017-09-13 Sprint to 2017-09-27 Sprint

#4 - 09/14/2017 08:41 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima

#5 - 09/27/2017 04:05 AM - Lucas Di Pentima

Updates at fd14dc21b - branch 12018-sync-groups-tool (WIP)

Added a first version of the new command arv-sync-groups, it currently does most of the described requirements, it's pending the reporting of untagged existing groups, and created group uuids.

About the first missing feature, I have a doubt: Should these groups be created as "system groups"? (ie: owner_uuid == {prefix}-tpzed-0000000000000). I ask because there's a unique index on groups that's composed of (owner_uuid, group_name) so to check for untagged but existing groups, I suppose I have to retrieve all groups from a particular owner and avoid name collision with other user's groups.

Related to the above, here's another question: Should this command use credentials from the environment vars as arv-put and others, or as it's more an admin tool, should read auth credentials from some config file?

#6 - 09/27/2017 04:05 AM - Lucas Di Pentima

- Status changed from New to In Progress

#7 - 09/27/2017 01:27 PM - Tom Clegg

How about making a single parent group, and using that as the owner_uuid of all synchronized groups?

- If --parent-group-uuid arg is given, use that uuid as the parent. Log a warning if it's not owned by system_user.
- Otherwise, create/find a group (owned by system-user) named "Externally synchronized groups", and use that as the parent

For credentials, just let the SDK do its default thing (env vars, then ~/.config/arvados/settings.conf).

#8 - 09/27/2017 06:21 PM - Lucas Di Pentima

- Target version changed from 2017-09-27 Sprint to 2017-10-11 Sprint

#9 - 10/11/2017 06:52 PM - Lucas Di Pentima

- Target version changed from 2017-10-11 Sprint to 2017-10-25 Sprint

#10 - 10/12/2017 05:17 PM - Lucas Di Pentima

First golang version at 46141b6c9098f30dcd6644845887789c1c9006da

It's basically a direct translation of the python version, I feel that I used too many type assertions, making the code "ugly", difficult to read. I suppose I should be creating specific types, for example a Set type instead of using map[string]struct{}

I was thinking about adding ListAll() to the go SDK, as it exist on PySDK arvados.util package. Do you think it would be useful?

I would love some feedback on golang code styling, any tips will be welcome. Thanks!

#11 - 10/12/2017 09:13 PM - Tom Clegg

You're right about all the type shenanigans. That map[string]arvadosclient.Dict stuff in the arvadosclient module wasn't the best idea. We can do something more like source:services/keep-balance/collection.go now -- we'll have to add Group and GroupList types in sdk/go/arvados (source:sdk/go/arvados/log.go is probably a good one to copy & modify) but once that's done, you can just load json right into a struct and use its fields like real fields, instead of all the string-keys and casting stuff.

SDK aside, you definitely shouldn't use arvadosclient.Dict for things like remoteGroups. Make a struct type with the fields you need to track, maybe something like

```
type groupInfo struct {
        Group           arvados.Group
        PreviousMembers map[string]bool
        CurrentMembers  map[string]bool
}
```

You can say fmt.Sprintf("%s", err), you don't need to say err.Error()

Golang style says error strings should start lowercase -- otherwise you end up chaining together into "Can't do this: Trouble at the mill: Shot off, completely."

I wouldn't bother trying to os.Stat() the input file to predict whether opening will work. Just open it, and if that doesn't work, report the error.

Instead of ("error reading csv file: %s", err) ... how about ("error reading %q: %s", *srcPath, err)

Appending to (and iterating over) a nil slice works, so you can probably drop some of the stuff like make([]interface{}, 0) -- just "var links []interface{}" would work here.

The "contains" function is unnecessary. You can do this in subtract():

```
if _, found := setB[element]; !found {
        result[element] = struct{}{}
}
```

Another way would be to use a map[string]bool instead of a map[string]struct{}, always assign true when you're assigning, and do this

```
if !setB[element] {
        result[element] = struct{}{}
}
```

#12 - 10/18/2017 05:29 PM - Lucas Di Pentima

Updates at ed6af9cb4

This commit is about tidying up the code, following the suggestions on note-11:

- Removed initial Python version.
- Removed superfluous input files checks and moved file opening to be before any API calls, to avoid doing them is there's a problem with the file.
- Added user, group & link types so they're populated by json decoding.
- Cleaned up ListAll() func so it can work with different resource types.
- Changed usage of set style types from map[string]struct{} to be map[string]bool to simplify membership checking.
- Corrected error messages to start with lowercase.
- Added more debug messages for -verbose mode.

#13 - 10/19/2017 01:10 PM - Lucas Di Pentima

Updates at ea10340803abade2d35212866fcbc1beb1acd533

- Added -parent-group-uuid parameter to specify a parent group for the remote groups. This group should be owned by the system user. If not provided, the tool will search for a group named "Externally synchronized groups" owned by the system user, or create one if not found.
- Skip (with a warning message) CSV lines with an empty field
- Check if current user is admin, fail otherwise.
- Use the parent group uuid when searching/creating remote groups.

Tests are still pending.

#14 - 10/19/2017 02:59 PM - Lucas Di Pentima

Found a bug, working on it.

#15 - 10/19/2017 08:24 PM - Tom Clegg

Functional

In the "evict" code, it seems like we should be removing user→group *and* group→user links, and we shouldn't be filtering on name=can_read

Arvados stuff

Should use arvados.ResourceListParams instead of arvadosclient.Dict for limit, offset, etc.

Should use arvados.User instead of a custom user type -- just need to add the Email field to source:sdk/go/arvados/user.go, and a UserList type similar to CollectionList in source:sdk/go/arvados/collection.go

I suspect the only resourceList interface you really need to implement ListAll is "Len() int"

In ListAll, simplify by always using limit=2^31-1 and terminate when len(response.Items)==0

Paging code should use order=uuid instead of the default modified_at -- less likely to miss groups in races that way

Shouldn't we be using group properties instead of tag links for "tagged as remote"? We can't yet filter on properties using the API filters, but the expected usage is for all role groups to be synchronized, so it might be best to filter on group_class==null and do additional filtering by properties on the client side for now.

Go style

Don't split lines just to stay in 80 chars (e.g., "error creating tag for group %q: %s")

#16 - 10/24/2017 12:15 PM - Lucas Di Pentima

Updates at a0dcf5e4b

- Manage two way links for group memberships.
- Simplify ListAll()
- resourceList interface simplification.
- Stop using arvadosclient in favor of arvados package for API server access.
- Do not tag remote groups with tag links. Use the parent group as a filter to search for remote groups.
- On group members loading, check that both membership links exist, just in case the command was interrupted after creating the first link for a user.
- User, UserList, Group & GroupList types go into Go SDK. (User type already existed, was expanded)

Pending:

- Tests

#17 - 10/24/2017 12:18 PM - Lucas Di Pentima

To be able to manually try this, I've symlinked my sdk/go/arvados dev copy to ~/go/src/git.curoverse.com/arvados.git/sdk/go/arvados

#18 - 10/24/2017 07:12 PM - Peter Amstutz

As a general comment, doMain() is 340 lines long, which makes it a bit hard to follow. It would be helpful if it was pulled apart into 3-4 smaller subroutines.

#19 - 10/25/2017 01:56 AM - Lucas Di Pentima

I've broken down the code into several funcs - e2c27eaae38a904a4b05d800affdc7860ee24e79
Continuing with the tests.

#20 - 10/25/2017 02:29 PM - Peter Amstutz

- The main loop that reads the CSV file should also go in its own function.
- The permission is "can_manage" not just "manage"

However I don't think should be granting "can_manage", because that means any member of the group can add and remove other members, which doesn't make sense since it is externally managed. It should be "can_write", which will allow writes to projects which are shared with the group for writing.

However, I think it would be even better to have a 3rd column that specifies "read" or "write" and update permission link accordingly. (The code will need to take into account that a user could transition between read and write permission on a given group).

#21 - 10/25/2017 03:25 PM - Lucas Di Pentima

Updates at f38cea6fa

- Replaced "manage" with "can_write" permission links

- Separated CSV file processing code into its own function
- Additional code splitting to be able to test it easily
- First couple of basic tests in place

Pending: write tests mocking arvados client calls

#22 - 10/25/2017 06:46 PM - Lucas Di Pentima

- Target version changed from 2017-10-25 Sprint to 2017-11-08 Sprint

#23 - 10/30/2017 02:40 PM - Lucas Di Pentima

Updates at 7327a28eb

- Membership removal fix
- Changed -path <input-file> to be a positional argument
- Enhanced help message
- Fixed group creation by adding them as role group class (they now appear on the Sharing tab)
- Added several tests

#24 - 10/30/2017 07:00 PM - Peter Amstutz

- The "Link" struct should be added to the "arvados" package.

The signature to ProcessFile() is 247 characters long. It should be broken up over multiple lines for readability.

I deliberately gave it a broken CSV file:

```
external_group,a@a
external_group,b@b
"another group,b@b
```

The error is very unhelpful:

```
2017/10/30 18:38:27 Group sync starting. Using "email" as users id and parent group UUID "
2tlax-j7d0g-djxpyba3fu6hug4"
2017/10/30 18:38:27 Found 4 users
2017/10/30 18:38:27 Found 3 remote groups
2017/10/30 18:38:27 error reading "blah.csv": %!s(<nil>)
```

Perhaps it could keep a count of lines so at least it can say "parse error on line X".

It looks like it could attempt to create multiple instances of the group→user can_read Link record, if there was an error between the two CreateLink calls. You should either

- refactor so that doesn't happen (record the existence of "group→user" and "group→user" links separately instead of a single "membership flag)
- or confirm that creating multiple links is harmless (I think this is true)

#25 - 10/30/2017 10:50 PM - Lucas Di Pentima

Updates at eb772d62e

- Link & LinkList types moved to Go SDK
- Enhanced error message when having a parsing error.
- Added test tear down function that cleans up all the links & remote groups created by every test.
- Added test that proves that records with empty fields are skipped

Regarding possible duplicated links, as far as I understand, they're harmless and will be deleted when removing the membership.

#26 - 10/31/2017 02:19 AM - Lucas Di Pentima

Updates at 9b83c3ee1

- Enhanced readability of ProcessFile() function.
- Added test using usernames instead of emails as users indentifiers.

#27 - 10/31/2017 08:00 PM - Peter Amstutz

For the tests, you should call arvadostest.StartAPI() instead of assuming the environment.

Instead of adding a bunch of test code to remove groups, consider using the database reset endpoint (this resets the database contents to fixtures). From run_test_server.py:

```
    httpclient.request(
```

```
        'https://{}/database/reset'.format(existing_api_host),
        'POST',
        headers={'Authorization': 'OAuth2 {}'.format(token)})
```

#28 - 10/31/2017 09:11 PM - Lucas Di Pentima

Updates at:8fc7e0dd5

- Added API server start/stop calls to test suite.
- Removed manual database cleanup in favour of using the database reset endpoint after every test run.
- Moved tool config set up code from the suite set up call to the test set up, because the database get reset to fixture state after every test run.

#29 - 11/02/2017 02:42 PM - Peter Amstutz

This LGTM @ 8fc7e0dd5d214e3881b8a56669f82d76aa70bfdb

#30 - 11/02/2017 03:05 PM - Anonymous

- Status changed from In Progress to Resolved

Applied in changeset arvados|commit:cc6f86f15e0d187cc1c84b874be3d2da7b20d19f.

#31 - 11/22/2017 08:47 PM - Tom Morris

Where is this tool packaged? Documented?

As an aside, it's too late now, but when did we switch from using Python for our client command line tools? I thought we were trying to reduce language diversity in the CLI tools. If we're going to make major changes like this, we should have an explicit decision about them.

#32 - 11/23/2017 09:02 PM - Lucas Di Pentima

Added documentation at 4b16a16d0 - branch 12018-tool-docs

#33 - 11/24/2017 04:58 PM - Lucas Di Pentima

- Status changed from Resolved to In Progress

- Target version changed from 2017-11-08 Sprint to 2017-12-06 Sprint

#34 - 11/24/2017 05:25 PM - Lucas Di Pentima

Updates at 53a6cafcd

HTML entities fixes

#35 - 12/04/2017 08:08 PM - Lucas Di Pentima

- Status changed from In Progress to Resolved