Arvados - Story #3699

[SDKs] Copy a pipeline instance, along with its input and output data, from one arvados instance to another

08/26/2014 03:29 PM - Tom Clegg

Status:	Resolved	**Start date:**	08/29/2014
Priority:	Normal	**Due date:**	
Assigned To:	Tim Pierce	**% Done:**	100%
Category:	SDKs	**Estimated time:**	0.00 hour
Target version:	2014-10-29 sprint		

Description

Use case: user can copy a pipeline instance between Arvados instances, in order to rerun a pipeline on another cluster and compare results with the original computation. Example:

1. User runs arv-copy 1h9kt-pipeline-uuid 1h9kt 4xphq to copy instance 1h9kt-pipeline-uuid to cluster 4xphq
2. User views the new pipeline instance on 4xphq's workbench
3. User clicks "run" on the copied pipeline template page (selecting an appropriate input collection, probably the input collection that was copied along with the pipeline instance and template)
4. Jobs run.
5. User uses "compare pipelines" on 4xphq to compare the original, copied 1h9kt pipeline instance with the new 4xphq instance that was just generated.

Syntax:

```
$ arv-copy [--recursive=true/false] [pipeline-instance-uuid] [source-arvados] [destination-arvados
]
```

By default, arv-copy exports the specified pipeline instance from the *source-arvados* instance and imports it to *destination-arvados*. arv-copy makes the following changes to the pipeline instance before importing it:

- renames uuid to properties.copied_from_pipeline_instance_uuid
- removes owner_uuid

The --recursive option, which defaults to true, also copies the following data:

- collections (copy blocks and then copy manifest_text)
  - Finding collections to copy: For each component in pipeline_instance.components(), append component.job().dependencies()
- docker images (collection copy + docker specific tags)
  - Copy docker images identified by collection hash in docker_image_locator
- pipeline templates (copy name, components)
- git repository (clone entire repository; update name of repository to use in components of target pipeline template)

If --recursive=false, copy only the pipeline instance, but emit a warning that the user will have to fix the pipeline template UUID by hand.

arv-copy returns an error if *pipeline-instance-uuid* refers to an object that cannot be copied between instances. For this story, arv-copy is only guaranteed to work on pipeline instance UUIDs. Future stories may expand this feature.

A warning is issued if arv-copy is asked to copy a pipeline instance in which:

- one or more components includes runtime_dependencies with a docker_image field, which is a symbolic name for a Docker image
- one or more components uses symbolic names for git revisions (e.g. a branch name, "master", etc)

For copying git commits, it is critical that we preserve the commit hashes between repositories, which means copying the commit history. This stackoverflow will probably provide important guidance: http://stackoverflow.com/questions/1365541/how-to-move-files-from-one-git-repo-to-another-not-a-clone-preserving-history.

Subtasks:

Task # 3742: arv-copy works on collections	**Resolved**

Task # 3744: arv-copy works on pipeline instances	**Resolved**
Task # 3743: arv-copy works on docker images	**Resolved**
Task # 3758: arv-copy works on git repos	**Resolved**
Task # 3838: Review 3699-arv-copy	**Resolved**
Task # 3784: arv-copy works on pipeline templates	**Resolved**
Task # 3759: arv-copy authenticates to multiple Arvados instances	**Resolved**

Associated revisions

Revision 35ade8a0 - 10/24/2014 05:20 PM - Tim Pierce

Merge branch '3699-arv-copy'

Closes #3699.

Revision ef56ac56 - 10/24/2014 05:58 PM - Tim Pierce

Merge branch '3699-arv-copy'

Refs #3699.

History

#1 - 08/26/2014 03:44 PM - Ward Vandewege

- Story points set to 2.0

#2 - 08/27/2014 02:06 PM - Ward Vandewege

- Target version changed from Arvados Future Sprints to 2014-09-17 sprint

#3 - 08/27/2014 03:57 PM - Peter Amstutz

- Description updated

#4 - 08/27/2014 04:09 PM - Peter Amstutz

- Story points changed from 2.0 to 3.0

#5 - 08/27/2014 04:12 PM - Peter Amstutz

- Description updated

#6 - 08/27/2014 04:20 PM - Tim Pierce

- Assigned To set to Tim Pierce

#7 - 08/27/2014 04:21 PM - Tim Pierce

- Subject changed from [SDKs] Copy objects from one arvados instance to another to [SDKs] A pipeline from one Arvados instance can be run on another instance

#8 - 08/27/2014 04:22 PM - Tom Clegg

- Subject changed from [SDKs] A pipeline from one Arvados instance can be run on another instance to [SDKs] Copy a pipeline instance, along with its input and output data, from one arvados instance to another

#9 - 08/27/2014 04:23 PM - Tom Clegg

- Description updated

#10 - 08/28/2014 02:53 PM - Tim Pierce

- Description updated

- Category set to SDKs

#11 - 08/29/2014 10:25 AM - Tim Pierce

- Description updated

#12 - 08/29/2014 10:33 AM - Tim Pierce

- Description updated

#13 - 08/29/2014 10:43 AM - Tim Pierce

- Description updated

#14 - 08/29/2014 11:59 AM - Tim Pierce

- Description updated

#15 - 08/29/2014 05:42 PM - Tim Pierce

- Description updated

#16 - 09/03/2014 02:38 PM - Peter Amstutz

In-progress review 1f3035c

1. Consider implementing part of the 'arvados.command' module (similarly to arv-put) and/or putting as much functionality into the SDK as possible.
2. Especially would like to see api_for_instance() in the SDK
3. I think Tom said that the uuid type map is available in the discovery document?
4. Want a --project-uuid option to specify the destination project (i.e. owner_uuid). We could get clever and determine the source and destination instances based on the instance portion of the uuid.
5. Want to log each thing copied along with the new UUID on the destination system.

#17 - 09/03/2014 02:52 PM - Tim Pierce

Peter Amstutz wrote:

> In-progress review 1f3035c
>
> 1. Consider implementing part of the 'arvados.command' module (similarly to arv-put) and/or putting as much functionality into the SDK as possible.
> 2. Especially would like to see api_for_instance() in the SDK

I'd like to do this too -- the immediate goal of course is to have arv-copy work on the command line, but to the extent I can structure this to be a wrapper around a sensible set of SDK functions, I will.

We could extend arvados.api() to take an "instance" or "config_file" argument -- e.g. arvados.api('v1', instance='qr1hi') would load configuration settings from ~/.config/arvados/qr1hi.conf.

> 1. I think Tom said that the uuid type map is available in the discovery document?

He did say this. I did not find it there. I'll try again.

> 1. Want a --project-uuid option to specify the destination project (i.e. owner_uuid). We could get clever and determine the source and destination instances based on the instance portion of the uuid.

A --project-uuid option is a good idea, but even if we guess the instance from the uuid, we'll still need to find the appropriate API token for it. I'm not sure that buys us anything useful.

> 1. Want to log each thing copied along with the new UUID on the destination system.

Can do.

#18 - 09/03/2014 02:52 PM - Tim Pierce

- Status changed from New to In Progress

#19 - 09/03/2014 03:32 PM - Tim Pierce

Tim Pierce wrote:

> Peter Amstutz wrote:
>
> > 1. I think Tom said that the uuid type map is available in the discovery document?
>
> He did say this. I did not find it there. I'll try again.

Okay, found it:

```
apischema = src._schema.schemas
```

```
obj_class = [k for k in apischema if apischema[k].get('uuidPrefix') == 'j7d0g']
if obj_class:
    return obj_class[0]
```

#20 - 09/11/2014 09:49 AM - Peter Amstutz

1. Add 'copy' subcommand to 'arv' frontend.
2. Give a friendlier error message, preferably explaining that the user needs to create a file and directing the user to the "manage_account" page to get the credentials:

```
$ arv-copy 4n8aq-d1hrv-51w0b47yd8hnt05 4n8aq 4xphq
Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 64, in main
    src_arv = api_for_instance(args.source_arvados)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 95, in api_for_instance
    cfg = arvados.config.load(config_file)
  File "/home/peter/work/arvados/sdk/python/arvados/config.py", line 31, in load
    with open(config_file, "r") as f:
IOError: [Errno 2] No such file or directory: '/home/peter/.config/arvados/4n8aq.conf'
```

3. We ought to allow users to paste this text as-is from the "manage_account" page (this confusion with settings.conf has been reported by actual users):

```
HISTIGNORE=$HISTIGNORE:'export ARVADOS_API_TOKEN=*'
export ARVADOS_API_TOKEN=3h3xxxxxxxxxxxxxxxxxxxxxxxxxr8jra3eabb
export ARVADOS_API_HOST=localhost:3001
export ARVADOS_API_HOST_INSECURE=true
```

4. There should probably be a way to specify an alternate search directory instead of only looking in '$HOME/.config/arvados' for the case of automated jobs that don't have a home directory.
5. This shouldn't fail: (trying to copy a pipeline instance)

```
$ arv-copy 4n8aq-d1hrv-51w0b47yd8hnt05 4n8aq 4xphq
Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 76, in main
    src=src_arv, dst=dst_arv)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 185, in copy_pipeline_instance
    for dep in job['dependencies']:
TypeError: 'NoneType' object is not iterable
```

6. Next I tried copying a pipeline template. Did this succeed? It doesn't appear to have copied the collections or git repository.

```
$ arv-copy 4n8aq-p5p6p-myx6p0vq84irkes 4n8aq 4xphq
{u'kind': u'arvados#pipelineTemplate', u'uuid': u'4xphq-p5p6p-l84js0szbtttuxk
', u'modified_at': u'2014-09-11T13:27:06Z', u'created_at': u'2014-09-11T13:27:06Z', u'description': None,
u'modified_by_client_uuid': u'4xphq-ozdt8-7sfww9tghj44cc3', u'owner_uuid': u'4xphq-tpzed-d6gnynp5uioqnxo
', u'href': u'/pipeline_templates/4xphq-p5p6p-l84js0szbtttuxk', u'etag': u'110ka4zc55m2uch7qeapvu27h', u'c
omponents': {u'hasher2': {u'nondeterministic': True, u'repository': u'peter2', u'script': u'hash', u'scrip
t_parameters': {u'input': {u'output_of': u'hasher'}}, u'runtime_constraints': {}, u'output_name': u'funky
hash man', u'script_version': u'fc45cbfa6fa0d33f7304b5c86a96449b52a68976'}, u'hasher': {u'nondeterministic
': True, u'repository': u'peter2', u'script': u'hash', u'script_parameters': {u'input': u'1235f41348b10eaf
f7d622dba7bd4a9f+83'}, u'runtime_constraints': {}, u'output_name': False, u'script_version': u'fc45cbfa6fa
0d33f7304b5c86a96449b52a68976'}}, u'modified_by_user_uuid': u'4xphq-tpzed-d6gnynp5uioqnxo
', u'name': u'hash copy'}
```

7. Next I tried --recursive on the same pipeline template. You need to add a "ensure_unique_name=true" to the create() call.

```
$ arv-copy --recursive 4n8aq-p5p6p-myx6p0vq84irkes 4n8aq 4xphq
Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 78, in main
    result = copy_pipeline_template(args.object_uuid, src=src_arv, dst=dst_arv)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 247, in copy_pipeline_template
    return dst.pipeline_templates().create(body=old_pt).execute()
```

```
  File "/usr/local/lib/python2.7/dist-packages/oauth2client/util.py", line 132, in positional_wrapper
    return wrapped(*args, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/apiclient/http.py", line 723, in execute
    raise HttpError(resp, content, uri=self.uri)
arvados.errors.ApiError: <HttpError 422 when requesting https://4xphq.arvadosapi.com/arvados/v1/pipeline_t
emplates?alt=json returned "#<PG::UniqueViolation: ERROR:  duplicate key value violates unique constraint
"pipeline_template_owner_uuid_name_unique"
DETAIL:  Key (owner_uuid, name)=(4xphq-tpzed-d6gnynp5uioqnxo, hash copy) already exists.
>">
```

8. I tried copying a collection. Finally something works? Hard to tell in between the debug spew.

```
$ arv-copy 4n8aq-4zz18-0qoogug1wpfu51o 4n8aq 4xphq
2014-09-11 09:37:22 arvados.arv-copy[12460] DEBUG: copying block db937a55ffd607b7a2238220bed2b0c8+71+Af8e9
1662692c162d9ff18e126d9e679dd90003b1@54241a92
DEBUG:arvados.arv-copy:copying block db937a55ffd607b7a2238220bed2b0c8+71+Af8e91662692c162d9ff18e126d9e679d
d90003b1@54241a92
2014-09-11 09:37:22 arvados.arv-copy[12460] INFO: Retrieved 71 bytes
INFO:arvados.arv-copy:Retrieved 71 bytes
2014-09-11 09:37:23 arvados.arv-copy[12460] DEBUG: saving 4n8aq-4zz18-0qoogug1wpfu51o
 manifest: . db937a55ffd607b7a2238220bed2b0c8+71+Af8e91662692c162d9ff18e126d9e679dd90003b1@54241a92 0:71:m
d5sum.txt

DEBUG:arvados.arv-copy:saving 4n8aq-4zz18-0qoogug1wpfu51o
 manifest: . db937a55ffd607b7a2238220bed2b0c8+71+Af8e91662692c162d9ff18e126d9e679dd90003b1@54241a92 0:71:m
d5sum.txt

{u'kind': u'arvados#collection', u'uuid': u'4xphq-4zz18-le36ua0jiuggyq7
', u'modified_at': u'2014-09-11T13:37:23Z', u'created_at': u'2014-09-11T13:37:23Z', u'description': None,
u'modified_by_client_uuid': u'4xphq-ozdt8-7sfww9tghj44cc3
', u'manifest_text': u'. db937a55ffd607b7a2238220bed2b0c8+71+A4ac65be7f7119422ccbd422c120cecf03b013fc2@542
41a93 0:71:md5sum.txt\n', u'owner_uuid': u'4xphq-tpzed-d6gnynp5uioqnxo
', u'properties': None, u'portable_data_hash': u'9c4ad41a0f62aeafdf95f4df222a251b+54', u'href': u'/collect
ions/4xphq-4zz18-le36ua0jiuggyq7', u'etag': u'aj1zyfo65uf8w39g22a79aprt', u'modified_by_user_uuid': u'
4xphq-tpzed-d6gnynp5uioqnxo', u'name': None}
```

9. On further inspection, the collection name doesn't appear to be copied. (remember to use "ensure_unique_name=true" to the create() call)

#21 - 09/15/2014 02:51 PM - Tim Pierce

New revision at 963c9e8. There has been a lot of diff churn here, and merges from master, (mostly around making sure that the names of repos and collections are changed in consistent ways when copying instances recursively) so you will probably find it easiest to diff master...HEAD from scratch.

Updates:

- More careful about copying collections to dst, making sure not to re-fetch blocks if the collection exists at the destination
- Copies all collections that match the regex for a collection UUID or hash, anywhere in the source object
- Copying git repositories: pushes explicitly to a branch named for the source git URL (e.g. git push dst git_git_qr1hi_arvadosapi_com_twp_git).
- More explicit about success or failure at the end
- Less debug output

The git commands still spit to stdout -- we can try to do something to make that quieter, but I found it useful since the git repo copying was one of the most annoying things to debug. Overall the verbosity of the command is diminished a lot.

#22 - 09/16/2014 10:43 AM - Peter Amstutz

1. Not addressed
2. Not addressed
3. Not addressed
4. Not addressed

I'm copying from my local development instance, so the configuration is probably broken, but this error messages is totally unhelpful in telling me what went wrong:

```
2014-09-16 09:25:01 arvados.arv-copy[26729] DEBUG: src_git_url: git@git.4n8aq.arvadosapi.com:peter2.git
Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 78, in main
    recursive=args.recursive)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 151, in copy_pipeline_instance
    recursive=True)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 199, in copy_pipeline_template
```

```
    copy_git_repos(pt, src, dst, dst_git_repo)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 249, in copy_git_repos
    copy_git_repo(repo, src, dst, dst_repo)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 348, in copy_git_repo
    .format(dst_git_repo, r['items_available']))
Exception: cannot identify source repo None; 0 repos found
```

- On further investigation there is a copy-and-paste error, it says "source" in both Exceptions in copy_git_repos().
- It appears that '--dst-git-repo' does not have a default value (such as choosing the first writable git repo in the destination list) so it is actually required on the command line, but this is not enforced.
- With some tinkering, I was able to copy a pipeline instance successfully. However, while arv-copy correctly updated the 'repository' portion of the component, it did not update the 'script_version' to point to the appropriate branch.

Next, I tried to copy another pipeline with associated collections:

```
$ arv-copy --dst-git-repo peter 4n8aq-d1hrv-0pxhk17pu24ac5j 4n8aq 4xphq
2014-09-16 10:03:18 arvados.arv-copy[28644] DEBUG: copying block 752192751a8a72ae6ac8b0fdb58d37df+5000+Ab06080
5321f7358d04a045e537c1a8631d16e700@542ab826
DEBUG:arvados.arv-copy:copying block 752192751a8a72ae6ac8b0fdb58d37df+5000+Ab060805321f7358d04a045e537c1a8631d
16e700@542ab826
2014-09-16 10:03:18 arvados.arv-copy[28644] INFO: Retrieved 5000 bytes
INFO:arvados.arv-copy:Retrieved 5000 bytes
2014-09-16 10:03:19 arvados.arv-copy[28644] DEBUG: saving 1235f41348b10eaff7d622dba7bd4a9f+83 manifest: . 7521
92751a8a72ae6ac8b0fdb58d37df+5000+Ab060805321f7358d04a045e537c1a8631d16e700@542ab826 0:5000:
4n8aq-8i9sb-c1utvb26t29nbq4.log.txt

DEBUG:arvados.arv-copy:saving 1235f41348b10eaff7d622dba7bd4a9f+83 manifest: . 752192751a8a72ae6ac8b0fdb58d37df
+5000+Ab060805321f7358d04a045e537c1a8631d16e700@542ab826 0:5000:4n8aq-8i9sb-c1utvb26t29nbq4.log.txt

Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 78, in main
    recursive=args.recursive)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 151, in copy_pipeline_instance
    recursive=True)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 198, in copy_pipeline_template
    pt = copy_collections(pt, src, dst)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in copy_collections
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in copy_collections
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in copy_collections
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in copy_collections
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 224, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 219, in copy_collections
    newc = copy_collection(obj, src, dst)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 314, in copy_collection
    del c['owner_uuid']
KeyError: 'owner_uuid'
```

- On investigation, this appears to be due to copying collections by content hash (which does not return 'owner_uuid') instead of by uuid.

#23 - 09/16/2014 11:08 AM - Tim Pierce

Peter Amstutz wrote:

> 1. Not addressed

34aac296 doesn't address this? I'm confused.

> 2. Not addressed

Adding a better error text for a missing arvados conf file, stand by.

3. Not addressed

I'm sympathetic to the confusion, but open to ideas about how we could effectively use the cut-and-pasted text. The core problem here is that arv-copy is the first tool that needs to know how to authenticate to multiple Arvados instances simultaneously, so the environment variables simply aren't going to be useful.

If the user cut-and-pastes the authentication environment variables from a source Arvados, and then cut-and-pastes the auth variables from the destination, which takes precedence?

4. Not addressed

Open to ideas here as well. How about permitting both --src=qr1hi --dst=4xphq and --src=$HOME/my-qr1hi.conf --dst=$HOME/4xphq.txt? i.e. if the src or dst arvados names start with a slash, treat that as an absolute path.

> I'm copying from my local development instance, so the configuration is probably broken, but this error messages is totally unhelpful in telling me what went wrong:
> [...]
>
> - On further investigation there is a copy-and-paste error, it says "source" in both Exceptions in copy_git_repos().
> - It appears that '--dst-git-repo' does not have a default value (such as choosing the first writable git repo in the destination list) so it is actually required on the command line, but this is not enforced.
> - With some tinkering, I was able to copy a pipeline instance successfully. However, while arv-copy correctly updated the 'repository' portion of the component, it did not update the 'script_version' to point to the appropriate branch.

Correct: --dst-git-repo is required. I'll have the Python argparse enforce it on the command line.

I investigated ways to identify a default git repo for the destination but didn't find any I liked. I'd prefer "choose a git repo randomly from ones you own" but repository ownership doesn't seem well defined. I'm uneasy about "choose a writable git repo randomly" if it means that you end up stuffing a lot of copied repository data into someone else's repository that you've been given write access to.

> Next, I tried to copy another pipeline with associated collections:
> [...]
>
> - On investigation, this appears to be due to copying collections by content hash (which does not return 'owner_uuid') instead of by uuid.

Ah, I slipped in the del c['owner_uuid'] before submitting but didn't test it on that particular case -- I didn't realize that collections could be returned without an owner_uuid. Fixing.

#24 - 09/16/2014 11:25 AM - Tim Pierce

Peter Amstutz wrote:

> - With some tinkering, I was able to copy a pipeline instance successfully. However, while arv-copy correctly updated the 'repository' portion of the component, it did not update the 'script_version' to point to the appropriate branch.

So the script_version needs to be updated even though the specified commit hash exists in both repositories under the same name?

(On reflection, that is obviously going to be true when script_version references a branch on the source. Ugh. But is it appropriate to rename when script_version references a hash?)

#25 - 09/16/2014 11:36 AM - Tim Pierce

New rev at 04a4fa5

- Issue helpful error message when config file cannot be opened
- Require --dst-git-repo argument
- Allow collections without owner_uuid (i.e. when retrieved by data hash rather than uuid)
- Corrected "source"/"destination" error message in copy_git_repo

#26 - 09/17/2014 09:53 AM - Tim Pierce

New revision at 9cff4a0 renames symbolic names found in script_version and supplied_script_version fields to the commit hashes they resolve to.

#27 - 09/17/2014 11:00 AM - Peter Amstutz

- Default logging level is still "DEBUG", that needs to be changed
- It should list which collections are being copied and "bytes uploaded" progress.

- Collections are copied without names which makes it impossible to figure out which is which
- Should be --project-uuid not --project_uuid
- I tried copying
  https://workbench.qr1hi.arvadosapi.com/pipeline_instances/qr1hi-d1hrv-f9wf1btyvsevep8
  and got
  https://workbench.9tee4.arvadosapi.com/pipeline_instances/9tee4-d1hrv-a0ecr7hb7sreu74
  then I did a clone and run, the result doesn't work:
  https://workbench.9tee4.arvadosapi.com/pipeline_instances/9tee4-d1hrv-72jjxiig9twrsio
  I suspect not all the collection data transferred over.

#28 - 09/17/2014 03:09 PM - Peter Amstutz

- Target version changed from 2014-09-17 sprint to 2014-10-08 sprint

#29 - 10/07/2014 03:37 PM - Tim Pierce

Peter Amstutz wrote:

> - Default logging level is still "DEBUG", that needs to be changed

Debug logging is now off by default. It can be enabled with a -v/--verbose flag.

> - It should list which collections are being copied and "bytes uploaded" progress.

I'll work on enabling a pretty progress report (like with arv-put) for non-verbose mode. For now, you get ugly line-by-line status reports with --verbose.

> - Collections are copied without names which makes it impossible to figure out which is which

It's not clear to me why this was happening (or whether it is still a problem). We are not removing the name field from the collection record when copying it to the destination. I'll try to reproduce.

> - Should be --project-uuid not --project_uuid

Fixed.

> - I tried copying
>   https://workbench.qr1hi.arvadosapi.com/pipeline_instances/qr1hi-d1hrv-f9wf1btyvsevep8
>   and got
>   https://workbench.9tee4.arvadosapi.com/pipeline_instances/9tee4-d1hrv-a0ecr7hb7sreu74
>   then I did a clone and run, the result doesn't work:
>   https://workbench.9tee4.arvadosapi.com/pipeline_instances/9tee4-d1hrv-72jjxiig9twrsio
>   I suspect not all the collection data transferred over.

Fixed a bug in collection copying (it was copying only the first block on each manifest line, oops). Added a --force flag, to copy collection blocks even if the collection record already exists at the destination (to fix broken collections like these).

Trivial pipelines can be copied and run. I'll try reproducing the broken pipelines you hit and see if they work now.

#30 - 10/07/2014 09:38 PM - Tim Pierce

Peter Amstutz wrote:

> - Collections are copied without names which makes it impossible to figure out which is which

I haven't been able to reproduce this problem, and copying collections now does preserve the name (e.g. https://workbench.4xphq.arvadosapi.com/collections/4xphq-4zz18-c5yckkig86ln3zc which was just copied from qr1hi).

#31 - 10/08/2014 06:10 PM - Tim Pierce

- Target version changed from 2014-10-08 sprint to 2014-10-29 sprint

#32 - 10/08/2014 07:14 PM - Tim Pierce

- Story points changed from 3.0 to 1.0

#33 - 10/17/2014 03:16 PM - Tim Pierce

Ready for re-review at 41c5cc5: collection copying now gets you a nice report by default (can disable with --no-progress).

Since this has had master merged back into it, there's a huge amount of diff churn -- I recommend git difftool 9cff4a0..HEAD sdk/python/arvados/commands/copy.py to view changes to copy.py just since your last review (which is where almost all of the changes have gone since then).

#34 - 10/17/2014 07:20 PM - Tim Pierce

Try again at revision b81c434: copy_git_repo should now correctly handle different branches in the source repository, and resolve the script_versions appropriately in the copied pipeline. Successfully copied qr1hi-d1hrv-fr2cgdn3q50f4ae to 4xphq.

#35 - 10/17/2014 08:53 PM - Peter Amstutz

- Help text says "Copy a pipeline instance from one Arvados instance to another", but should mention you can copy templates, collections
- Needs to be smarter about finding content hashes in script_parameters. For example, 4xphq-d1hrv-t4soj38x1s5bbfs has a content hash embedded in a string: "$(file 3229739b505d2b878b62aed09895a55a+142/HWI-ST1027_129_D0THKACXX.1_1.fastq)"
- Use logger consistently, e.g. use logger.info() instead print >>sys.stderr
- Prefer isinstance(obj, basestr) instead of type(obj) in [str, unicode]
- ensure_unique_name is a parameter of the create method, not an object field. I think you want to be using dst.pipeline_instances().create(body=pi, ensure_unique_name=True).execute() (you should test this)
- Failed: arv-copy --verbose --src qr1hi --dst 4xphq --dst-git-repo peter qr1hi-d1hrv-44qifjjtuoh2xcw

```
2014-10-17 16:51:49 arvados.arv-copy[17845] DEBUG: src_git_url: git@git.qr1hi.arvadosapi.com:arvados.git
2014-10-17 16:51:49 arvados.arv-copy[17845] DEBUG: dst_git_push_url: git@git.4xphq.arvadosapi.com:peter.gi
t
Cloning into bare repository '/tmp/tmpQfF59g'...
remote: Counting objects: 47209, done.
remote: Compressing objects: 100% (17572/17572), done.
remote: Total 47209 (delta 33108), reused 38969 (delta 26834)
Receiving objects: 100% (47209/47209), 7.33 MiB | 3.01 MiB/s, done.
Resolving deltas: 100% (33108/33108), done.
Checking connectivity... done.
Everything up-to-date
2014-10-17 16:51:55 arvados.arv-copy[17845] DEBUG: Copying collection qr1hi-4zz18-0gobfjfihm0bi1p
2014-10-17 16:51:55 arvados.arv-copy[17845] DEBUG: Copying block 12f45b121fde0cc5a80656050c2a5acc (2992674
0 bytes)
qr1hi-4zz18-0gobfjfihm0bi1p: 0M / 28M 0.0%
2014-10-17 16:52:09 arvados.arv-copy[17845] DEBUG: saving qr1hi-4zz18-0gobfjfihm0bi1p
 manifest: . 12f45b121fde0cc5a80656050c2a5acc+29926740+Ab2a278ea3fdeaae02437bf153d3916b22c956b3c@5453f679
0:29926740:HWI-ST1027_129_D0THKACXX.1_1.sam

2014-10-17 16:52:09 arvados.arv-copy[17845] DEBUG: Copying collection qr1hi-4zz18-ah4fm98e4osc5ua
2014-10-17 16:52:09 arvados.arv-copy[17845] DEBUG: Copying block c313fb45a7d5f7580ba1eebb1e071b46 (2510438
4 bytes)
qr1hi-4zz18-ah4fm98e4osc5ua: 0M / 23M 0.0%
2014-10-17 16:52:22 arvados.arv-copy[17845] DEBUG: saving qr1hi-4zz18-ah4fm98e4osc5ua
 manifest: . c313fb45a7d5f7580ba1eebb1e071b46+25104384+A53a28ae3ebcc10f0ca106e38d785ce61623d4d26@5453f686
0:12552192:HWI-ST1027_129_D0THKACXX.1_1.fastq 12552192:12552192:HWI-ST1027_129_D0THKACXX.1_2.fastq

Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 103, in main
    args)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 196, in copy_pipeline_instance
    pi = copy_collections(pi, src, dst, args)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 272, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 267, in copy_collections
    newc = copy_collection(obj, src, dst, args)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 438, in copy_collection
```

```
    return dst.collections().create(body=c).execute()
  File "/usr/local/lib/python2.7/dist-packages/oauth2client/util.py", line 132, in positional_wrapper
    return wrapped(*args, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/apiclient/http.py", line 723, in execute
    raise HttpError(resp, content, uri=self.uri)
arvados.errors.ApiError: <HttpError 422 when requesting https://4xphq.arvadosapi.com/arvados/v1/collection
s?alt=json returned "Portable data hash does not match hash of manifest_text">
```

#36 - 10/20/2014 06:23 PM - Tim Pierce

Nice finds. New version at cdaf5c7.

Peter Amstutz wrote:

- Help text says "Copy a pipeline instance from one Arvados instance to another", but should mention you can copy templates, collections

Updated.

- Needs to be smarter about finding content hashes in script_parameters. For example, 4xphq-d1hrv-t4soj38x1s5bbfs has a content hash embedded in a string: "$(file 3229739b505d2b878b62aed09895a55a+142/HWI-ST1027_129_D0THKACXX.1_1.fastq)"

Ugh, good catch. Fixed, but it required substantially rewriting copy_collections stuff so that the command line arguments can be intelligently rewritten. (As a bonus, copy_collections now also keeps track of which collections have been copied in this session, so it can avoid repeatedly asking *dst* whether this collection exists.)

- Use logger consistently, e.g. use logger.info() instead print >>sys.stderr

Done.

- Prefer isinstance(obj, basestr) instead of type(obj) in [str, unicode]

Thanks for that. Done.

- ensure_unique_name is a parameter of the create method, not an object field. I think you want to be using dst.pipeline_instances().create(body=pi, ensure_unique_name=True).execute() (you should test this)

Tested and confirmed -- done.

- Failed: arv-copy --verbose --src qr1hi --dst 4xphq --dst-git-repo peter qr1hi-d1hrv-44qifjjtuoh2xcw
  [...]

The bug here was that the source manifest had no trailing newline, but arv-copy was mistakenly adding one in the destination manifest. Fixed.

#37 - 10/20/2014 06:34 PM - Peter Amstutz

```
$ arv-copy --verbose --src qr1hi --dst 4xphq --dst-git-repo peter qr1hi-d1hrv-44qifjjtuoh2xcw
2014-10-20 14:33:57 arvados.arv-copy[3159] DEBUG: src_git_url: git@git.qr1hi.arvadosapi.com:arvados.git
2014-10-20 14:33:57 arvados.arv-copy[3159] DEBUG: dst_git_push_url: git@git.4xphq.arvadosapi.com:peter.git
Cloning into bare repository '/tmp/tmpPmFdsM'...
remote: Counting objects: 47209, done.
remote: Compressing objects: 100% (17573/17573), done.
remote: Total 47209 (delta 33108), reused 38973 (delta 26833)
Receiving objects: 100% (47209/47209), 7.33 MiB | 4.14 MiB/s, done.
Resolving deltas: 100% (33108/33108), done.
Checking connectivity... done.
Everything up-to-date
Traceback (most recent call last):
  File "/home/peter/work/arvados/sdk/cli/bin/arv-copy", line 4, in <module>
    main()
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 107, in main
    args)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 200, in copy_pipeline_instance
    pi = copy_collections(pi, src, dst, args)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 294, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 294, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
```

```
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 294, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 294, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 294, in copy_collections
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 294, in <dictcomp>
    return {v: copy_collections(obj[v], src, dst, args) for v in obj}
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 291, in copy_collections
    obj = arvados.util.collection_uuid_pattern.sub(copy_collection_fn, obj)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 280, in copy_collection_fn
    dst_col = copy_collection(src_id, src, dst, args)
  File "/home/peter/work/arvados/sdk/python/arvados/commands/copy.py", line 387, in copy_collection
    c = src.collections().get(uuid=obj_uuid).execute()
  File "/usr/local/lib/python2.7/dist-packages/oauth2client/util.py", line 132, in positional_wrapper
    return wrapped(*args, **kwargs)
  File "/usr/local/lib/python2.7/dist-packages/apiclient/http.py", line 723, in execute
    raise HttpError(resp, content, uri=self.uri)
arvados.errors.ApiError: <HttpError 404 when requesting https://qr1hi.arvadosapi.com/arvados/v1/collections/%3
C_sre.SRE_Match%20object%20at%200x7ffed8ae5920%3E?alt=json returned "Path not found">
```

#38 - 10/20/2014 07:20 PM - Tim Pierce

Revision at 8bd432b:

The re.sub 'repl' function takes a MatchObject as argument, not a string. Oops.

Also we need to do manifest.splitlines(True) in order to be able to tell whether the manifest ends with a newline in the first place.

Used the code at this head to clone several arv-run pipelines successfully (the most complex pipelines I could find that could be copied in a reasonable amount of time for testing)

#39 - 10/20/2014 07:57 PM - Peter Amstutz

- I used arv-copy and got a pipeline: 4xphq-d1hrv-c25qsfov6u70jt1 It has been given an unhelpful generic name (granted the source pipeline name is null) and a spurious "None" in its description:

```
New pipeline instance
Pipeline copied from qr1hi-d1hrv-44qifjjtuoh2xcw

None
```

- I don't know if there's a good way to efficiently search pipeline templates on the destination, but after copying the same stuff over and over again I'm up to at least 6 duplicates of the same template: "Tutorial align using bwa mem copied from qr1hi-p5p6p-itzkwxblfermlwv (5)"

- I crashed it again. At this point, the configuration file for "peter" doesn't exist yet.

```
$ arv-copy  --src qr1hi --dst peter --dst-git-repo peter qr1hi-d1hrv-44qifjjtuoh2xcw
Traceback (most recent call last):
  File "/usr/lib/python2.7/logging/__init__.py", line 859, in emit
    msg = self.format(record)
  File "/usr/lib/python2.7/logging/__init__.py", line 732, in format
    return fmt.format(record)
  File "/usr/lib/python2.7/logging/__init__.py", line 471, in format
    record.message = record.getMessage()
  File "/usr/lib/python2.7/logging/__init__.py", line 335, in getMessage
    msg = msg % self.args
TypeError: not all arguments converted during string formatting
Logged from file copy.py, line 559
```

#40 - 10/20/2014 08:08 PM - Peter Amstutz

- This is not terribly reassuring:

```
qr1hi-4zz18-0gobfjfihm0bi1p: 0M / 28M 0.0%
b9edd3ac5dd0717f6ca587c3b2ec9885+83: 0M / 0M 0.0%
```

(presumably the blocks are actually being sent, but there's no extra print statement for 100% when it's done with a collection)

#41 - 10/20/2014 08:39 PM - Peter Amstutz

```
$ arv-copy  --src qr1hi --dst 4n8aq --dst-git-repo peter2 qr1hi-d1hrv-44qifjjtuoh2xcw
loning into bare repository '/tmp/tmpMIOc6_'...
remote: Counting objects: 47209, done.
remote: Compressing objects: 100% (17576/17576), done.
remote: Total 47209 (delta 33107), reused 38974 (delta 26830)
Receiving objects: 100% (47209/47209), 7.33 MiB | 4.58 MiB/s, done.
Resolving deltas: 100% (33107/33107), done.
Checking connectivity... done.
Total 0 (delta 0), reused 0 (delta 0)
To /home/peter/work/arvados_prod_repos/peter2
 * [new branch]      git_git_qr1hi_arvadosapi_com_arvados_git_3cc80b447efcaf416ea4d6857d6d40583e462ff8 -> git_
git_qr1hi_arvadosapi_com_arvados_git_3cc80b447efcaf416ea4d6857d6d40583e462ff8
qr1hi-4zz18-0gobfjfihm0bi1p: 0M / 28M 0.0%
b9edd3ac5dd0717f6ca587c3b2ec9885+83: 0M / 0M 0.0%
142e99c2dec346e621fd3eeb30a63387+1050: 1408M / 1442M 97.6%
2014-10-20 16:14:43 arvados.arv-copy[8356] INFO:
2014-10-20 16:14:43 arvados.arv-copy[8356] INFO: Success: created copy with uuid 4n8aq-d1hrv-1gswgofj4qon38c
```

It should have copied "qr1hi-4zz18-ah4fm98e4osc5ua" ("sample" in "script_parameters") but it was skipped for some reason. However, the fact that collections are showing up in script_parameters at all is a much bigger problem: #4269

#42 - 10/20/2014 08:40 PM - Tim Pierce

At commit 81c3241:

Peter Amstutz wrote:

- I used arv-copy and got a pipeline: 4xphq-d1hrv-c25qsfov6u70jt1 It has been given an unhelpful generic name (granted the source pipeline name is null) and a spurious "None" in its description:
  [...]

Fixed the "None" description (this just has to be a little more clever than pi.get('description', '')

I don't know if there's a better solution than an unhelpful generic name, when copying a pipeline that already doesn't have a name. Happy to take suggestions there :-)

- I don't know if there's a good way to efficiently search pipeline templates on the destination, but after copying the same stuff over and over again I'm up to at least 6 duplicates of the same template: "Tutorial align using bwa mem copied from qr1hi-p5p6p-itzkwxblfermlwv (5)"

This is based on explicit (albeit verbal) direction from Tom: if a user runs arv-copy twice on the same pipeline, arv-copy copies all of the source objects to the destination, period. Collections and git repositories are not duplicated because content-addressed storage essentially prevents it, but we are deliberately not attempting to be clever about reusing templates that have already been copied.

- I crashed it again. At this point, the configuration file for "peter" doesn't exist yet.
  [...]

In that case, it will abort anyway, but this is definitely a bug in the abort code. Fixed:

```
(arv3699)hitchcock:/home/twp/arvados/sdk/python% arv-copy --src qr1hi --dst zzzzzz --dst-git-repo twp
qr1hi-p5p6p-itzkwxblfermlwv
2014-10-20 16:26:44 arvados.arv-copy[1887] INFO: arv-copy: Could not open config file /home/twp/.config/arvado
s/zzzzzz.conf: [Errno 2] No such file or directory: '/home/twp/.config/arvados/zzzzzz.conf'
You must make sure that your configuration tokens
for Arvados instance zzzzzz are in /home/twp/.config/arvados/zzzzzz.conf and that this
file is readable.
```

(presumably the blocks are actually being sent, but there's no extra print statement for 100% when it's done with a collection)

Yeah, that does look alarming. Added a statement to finish out the progress report after the copy is done. (It still uses the actual bytes_written/bytes_expected numbers, so it actually will report less than 100% if those numbers don't match up for some reason.)

#43 - 10/21/2014 04:09 PM - Peter Amstutz

- File arv-copy-perf.png added

I'm copying a pipeline with a 5990M collection. I noticed this code:

```
                    data = src_keep.get(word)
                    dst_locator = dst_keep.put(data)
```

See attached image, there's a very clear falloff between blocks -- doing this sequentially isn't optimal. Download and upload could proceed concurrently. Also, I suspect we could get better utilization if we downloaded 2 blocks at a time. But in the interests of getting arv-copy out the door we probably shouldn't do anything about it now.

#44 - 10/21/2014 04:54 PM - Peter Amstutz

```
arv-copy --src qr1hi --dst 4n8aq --dst-git-repo peter2 qr1hi-d1hrv-xbcpup0o8hexwwn
```

Everything succeeded! Yay!

Went to re-run the pipeline locally:

```
Error creating job for component run_lobSTR: Docker image locator not found for bcosc/lobstr
```

Boo! It needs to either create docker_image_repo+tag links for the Docker image, or rewrite the docker_image field of runtime_constraints in the pipeline to use a Docker image hash or a Arvados collection hash.

#45 - 10/22/2014 05:19 PM - Tim Pierce

Peter Amstutz wrote:

> See attached image, there's a very clear falloff between blocks -- doing this sequentially isn't optimal. Download and upload could proceed concurrently. Also, I suspect we could get better utilization if we downloaded 2 blocks at a time. But in the interests of getting arv-copy out the door we probably shouldn't do anything about it now.

Agreed on both counts. I'll file a ticket to investigate implementing concurrency for arv-copy. arvados.keep.KeepClient already runs requests on threads internally -- it might be as simple as allowing the caller to tell KeepClient to return immediately and not wait for the request to finish?

> Boo! It needs to either create docker_image_repo+tag links for the Docker image, or rewrite the docker_image field of runtime_constraints in the pipeline to use a Docker image hash or a Arvados collection hash.

Good catch. I think we should do as little rewriting as possible -- the copied pipeline should be identical to the original whenever possible. I'll have it add create the Docker tag links.

#46 - 10/22/2014 08:36 PM - Tim Pierce

Peter Amstutz wrote:

> Boo! It needs to either create docker_image_repo+tag links for the Docker image, or rewrite the docker_image field of runtime_constraints in the pipeline to use a Docker image hash or a Arvados collection hash.

Now available: d3dbc2c includes copy_docker_images to copy any Docker images named by image+tag in the source pipeline, and create docker_image_repo+tag and docker_image_hash links at the destination. I think this solution is necessary: even if we rewrite the pipeline instance, the template will be unusable without docker image links.

What I'm not sure about is how to choose which link is the correct one, since apparently docker_image_repo+tag links do not enforce uniqueness on name, to wit:

```
>>> import arvados.commands.copy
>>> api = arvados.commands.copy.api_for_instance('qr1hi')
>>> api.links().list(filters=[
        ['link_class','=','docker_image_repo+tag'],
        ['name','=','arvados/jobs:latest']
    ]).execute()['items_available']
9
```

What should I be doing here?

#47 - 10/23/2014 05:43 PM - Tim Pierce

Okay: at e845145 this code uses arvados.commands.keepdocker.list_images_in_arv to identify which docker image to pull from the source. A pipeline copied with this code produces appropriate links for arv-keepdocker to find in the destination:

```
(arv3699)hitchcock:/home/twp/arvados/sdk/python% ARVADOS_API_HOST=4xphq.arvadosapi.com ARVADOS_API_TOKEN=*****
* arv-keepdocker | grep lobstr
bcosc/lobstr                    latest      ea32030ce02e  4xphq-4zz18-x72yrfegn8iwmmw
    Thu Oct 23 17:09:57 2014
```

Try again?

#48 - 10/24/2014 03:10 PM - Peter Amstutz

When you're copying a pipeline instance that's been run already, the docker image will have been resolved into an explicit keep locator in the docker_image_locator field. Arv-copy needs to copy that one and not just the latest image with the same name.

#49 - 10/24/2014 03:41 PM - Tim Pierce

Peter Amstutz wrote:

> When you're copying a pipeline instance that's been run already, the docker image will have been resolved into an explicit keep locator in the docker_image_locator field. Arv-copy needs to copy that one and not just the latest image with the same name.

That's a deliberate decision:

- The collection identified in the docker_image_locator field will be copied anyway by copy_collections.
- If the docker image+tag still resolves to that image when the pipeline is copied, the appropriate links will be added on the destination to make sure that is still true.
- If the docker image has been updated on the source, so that docker image+tag no longer resolves to the docker_image_locator named in the pipeline, arv-copy will copy both docker images -- the docker_image_locator and the new docker image+tag.

The objective is to make sure that the current state of the pipeline is copied as exactly as possible, including "what will happen if I rerun this pipeline with new options."

Let me know if you think this reasoning is unsound.

#50 - 10/24/2014 05:25 PM - Tim Pierce

- Status changed from In Progress to Resolved

- % Done changed from 86 to 100

Applied in changeset arvados|commit:35ade8a042094a27e2ca5cfd5e9754aa3513410c.

#51 - 10/24/2014 05:29 PM - Peter Amstutz

I transferred an entire pipeline instance from qr1hi and it ran on my workstation with no changes. Merge this thing!

Files

arv-copy-perf.png	25.8 KB	10/21/2014	Peter Amstutz