Arvados - Story #11942

[CWL] arvados-cwl-runner should support tagging output collection using properties

07/05/2017 08:41 PM - Tom Morris

Status:	Closed	**Start date:**	07/05/2017
Priority:	Normal	**Due date:**	
Assigned To:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Subtasks:

Task # 13846: Review 11942-tagging-support	**Closed**

Related issues:

Related to Arvados - Feature #8641: add key/value support for tags	**Closed**	**03/04/2016**
Related to Arvados - Feature #14016: [API] Container request can provide exis...	**New**	
Related to Arvados - Feature #17004: Can set output_properties on output coll...	**New**	

History

#1 - 07/05/2017 08:55 PM - Tom Morris

The current output collection tagging using Links, but when tag support is available in the Collection.properties field, arvados-cwl-runner should switch to using that instead.

#2 - 07/11/2017 06:14 PM - Tom Clegg

Use a new command line option for "tag via properties" so old code continues to work as before until we figure out the tag migration strategy.

#3 - 07/11/2017 06:21 PM - Tom Clegg

Example

```
arvados-cwl-runner --output-properties "foo:bar,baz:qux" ...
```

results in

```
"properties":{
  "foo":"bar",
  "baz":"qux"
}
```

#4 - 07/11/2017 06:26 PM - Lucas Di Pentima

- Story points set to 1.0

#5 - 03/14/2018 07:49 PM - Tom Morris

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

#6 - 04/18/2018 03:42 PM - Tom Morris

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

#8 - 07/18/2018 03:51 PM - Tom Morris

- Assigned To set to Fuad Muhic

- Target version changed from Arvados Future Sprints to 2018-08-01 Sprint

#9 - 07/19/2018 02:09 PM - Peter Amstutz

Suggestion, as an alternative or in addition to "--output-properties" there should be a command line flag that takes an input YAML file and merges that into the output collection properties.

#10 - 07/20/2018 12:12 PM - Fuad Muhic

- Status changed from New to In Progress

#11 - 07/27/2018 04:27 PM - Fuad Muhic

11942-tagging-support is ready for a review @ 7eadb7fa179c18e41066709e8645a1e6eaad655c
test-run: https://ci.curoverse.com/job/developer-run-tests/829/

- Added --output-properties and --output-properties-yaml to arvados-cwl-runner
- propertied defined by both flags are combined into into one dictionary (if conflict happens, properties defied by --output-properties are taken)

#12 - 07/30/2018 02:56 PM - Lucas Di Pentima

I've tried to test this by running a-c-r against both arvbox and 4xphq and although the argument is accepted by the command, the output collection properties don't include the requested tags.
Am I making some mistake trying it, the CWL file for testing is like this:

```
class: CommandLineTool
cwlVersion: v1.0
inputs: []
outputs: []
stdout: output.txt
baseCommand: [echo, "Hello world"]
```

The command against 4xphq:

```
$ arvados-cwl-runner --wait --disable-reuse --api=containers --output-properties "foo:bar,baz:qux" echo.cwl
```

The properties field on the output collection is like this:

```
[...]
  "properties": {
    "type": "output",
    "container_request": "4xphq-xvhdp-hhs2zenp91tc6vi"
  },
[...]
```

#13 - 08/01/2018 03:15 PM - Peter Amstutz

- Target version changed from 2018-08-01 Sprint to 2018-08-15 Sprint

#14 - 08/13/2018 03:07 PM - Peter Amstutz

OK, we made a mistake here.

There's several ways a-c-r can submit a container request and get a collection as output.

1. Running a in --local mode. When the workflow ends, it calls make_output_collection and can set properties.
2. Running in --submit --wait mode. When the workflow ends, it calls make_output_collection and can set properties, and sets the PDH of the output on the container record. The container request output creates a new collection with the same content, but does not copy the properties (because it is referenced by PDH, it could find multiple collection records with conflicting properties). If the client a-c-r is still waiting for a result, it could apply the properties to the container requests's output.
3. Running in --submit --no-wait mode. Same as above, except there's no client process available to apply the properties.
4. As a variation, when running a single CommandLineTool in --submit mode it only submits a single container request for the tool, there is no separate workflow runner container. If the submitter process blocks, it could set properties when after the container request is completed. If it does not block (--no-wait) again there is no process available to set properties on the final output collection.

As a result, to property implement this feature seems to require API support to make it possible to specify the properties ahead of time, such as #14016

#16 - 08/13/2018 06:06 PM - Tom Morris

- Related to Feature #14016: [API] Container request can provide existing collection UUID that will accept CR output added

#17 - 08/13/2018 06:58 PM - Tom Morris

What, specifically, is the mistake that we made? We only implemented support for some of the code paths? Does that also imply that the current support for name and expiration dates is only implemented for some of the cases?

Is the fact that there are so many different code paths a good and necessary thing, or something that we need to fix?

Is part of the problem that we record PDH instead of UUID for the output collection (and it it's ambiguous)? Should we fix that?

If we can enumerate what the problems/gaps are, we can formulate a plan to address them.

#18 - 08/13/2018 07:25 PM - Peter Amstutz

Tom Morris wrote:

> What, specifically, is the mistake that we made?

The mistake was putting the story on the sprint without realizing the assumptions had changed going from links/jobs API to properties/containers API.

> We only implemented support for some of the code paths? Does that also imply that the current support for name and expiration dates is only implemented for some of the cases?

Name and expiration dates are already handled by the API server as a special case (they are fields of the container request which are applied to the output collection). So one possible solution is to make properties a similar special case.

> Is the fact that there are so many different code paths a good and necessary thing, or something that we need to fix?

Good and necessary. The main takeaway should be that sometimes the client is still running after the container, and sometimes not, so we shouldn't rely on it to do a fixup right at the end.

> Is part of the problem that we record PDH instead of UUID for the output collection (and it it's ambiguous)? Should we fix that?

It complicates things because it interferes with the solution of defining that properties on the output collection are copied the container request output. We still want to record the PDH for provenance, but we could record the uuid in addition to the PDH (then there would be something to copy from.)

#19 - 08/13/2018 07:45 PM - Tom Morris

Peter Amstutz wrote:

> Name and expiration dates are already handled by the API server as a special case (they are fields of the container request which are applied to the output collection). So one possible solution is to make properties a similar special case.

Would it make sense to generalize this to "output_collection_fields." or some such, which is a hash/dict containing things which should be set on the output container and move name, expiration date, and properties there? This is similar to "store the entire collection record" idea which was mentioned in chat, but would be restricted to just the handful of supported fields. One advantage would be future extensibility and consistent implementation.

Whether it's a new generalized field or a new specific field just for properties, it seems to make sense to be consistent with the other two collection fields which are already implemented.

Designing an entirely new mechanism seems like overkill/over engineering.

#20 - 08/13/2018 08:32 PM - Peter Amstutz

A generalized field for setting output collection fields would be fine.

However, for about the same amount of work, I think #14016 (container request can specify desired output_uuid) make it possible to solve the immediate problem but could apply to a wider variety of circumstances.

I agree that a generalized "run some other user code after an event" feature isn't needed to solve this specific problem, although if one existed we wouldn't need to have this conversation.

#21 - 08/15/2018 03:47 PM - Tom Morris

- Target version changed from 2018-08-15 Sprint to Arvados Future Sprints

#22 - 12/18/2019 08:24 PM - Peter Amstutz

- Assigned To deleted (Fuad Muhic)

- Status changed from In Progress to New

#23 - 07/06/2021 09:15 PM - Peter Amstutz

- Target version deleted (Arvados Future Sprints)

#24 - 08/03/2021 05:20 PM - Peter Amstutz

- Related to Feature #17004: Can set output_properties on output collection of a container request added

#25 - 08/03/2021 07:01 PM - Peter Amstutz

- Status changed from New to Closed