

## Arvados - Bug #11097

### [API] Reuse containers even when multiple matching containers exist with differing outputs

02/13/2017 04:39 PM - Tom Clegg

<b>Status:</b>	Resolved	<b>Start date:</b>	02/13/2017
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Tom Clegg	<b>% Done:</b>	100%
<b>Category:</b>	API	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2017-03-01 sprint		

#### Description

### Background

Sometimes, running the same container twice on the same inputs can result in two successes with two different outputs. This can mean a number of things, including

- undetected failure in one or both cases, perhaps resulting in bogus output
- both outputs are correct, but have non-meaningful differences (like an "output produced at {timestamp}" comment in an output file)

The second case is common in practice.

Currently, the API server disables the container re-use logic entirely when it detects that two re-use candidates produced different outputs. This causes the following undesirable pattern:

1. Run container "X" as part of a workflow w1
2. Re-use container "X" automatically in subsequent workflows w2..w5, saving time
3. Run workflow w4 with re-use disabled, e.g., to get runtime stats or verify reproducibility -- this runs container "X1" which is identical to "X" but produces different (but still correct) output
4. Run workflow w5..w9 with re-use enabled
5. Oops, even when re-running workflow w5, container "X" is not eligible for reuse ever again, because "X1" exists.

### Desired behavior

Use the oldest matching container whose output and log collections exist, aren't trashed, and are readable by the current user.

If we used the newest matching container, we would have the following problem:

1. Run container X, producing out1
2. Run workflows w1..w9 that reuse X and do a lot of downstream work on out1
3. Re-run workflows w1..w9 → lots of reused containers
4. Re-run container X1, producing out2
5. Re-run workflows w1..w9 → arvados chooses X1 now, so all downstream work has to be redone

Using the oldest matching container fixes the problems given above, while admitting the converse problem:

1. Run container "X"
2. Notice that container "X" exited 0 but produced bogus output because of a bug in the container process or Arvados itself
3. Run container again with re-use disabled: "X1" produces correct output
4. Run a workflow that makes use of this container
5. Oops, the workflow gets the bogus "X" output instead of the newer "X1" output

This is the lesser evil in that re-running the same container -- i.e., without fixing the underlying problem that allowed it to exit 0 with bogus output -- is not a viable solution anyway.

### Implementation

Disable this check in <source:services/api/app/models/container.rb>

```
if outputs.count.count != 1
  Rails.logger.debug("Found #{outputs.count.length} different outputs")
```

**Subtasks:**

Task # 11140: Update tests

**Resolved**

Task # 11111: Review 11097-reuse-impure

**Resolved****Associated revisions****Revision 0c529ed0 - 02/22/2017 06:19 PM - Tom Clegg**

Merge branch '11097-reuse-impure'

closes #11097

**History****#1 - 02/13/2017 04:40 PM - Tom Clegg***- Description updated***#2 - 02/13/2017 04:40 PM - Tom Clegg***- Description updated***#3 - 02/13/2017 04:43 PM - Tom Clegg***- Description updated***#4 - 02/13/2017 06:45 PM - Tom Clegg**11097-reuse-impure @ [264ffa31bae106bb6c36643e13186289b6cd0e18](#)

...fails a few tests -- but perhaps only because it changes the behavior as intended.

**#5 - 02/13/2017 06:45 PM - Tom Clegg***- Status changed from New to In Progress***#6 - 02/13/2017 06:45 PM - Tom Clegg***- Target version set to 2017-02-15 sprint***#7 - 02/15/2017 08:08 PM - Tom Clegg***- Target version changed from 2017-02-15 sprint to Arvados Future Sprints***#8 - 02/15/2017 09:33 PM - Tom Clegg***- Target version changed from Arvados Future Sprints to 2017-03-01 sprint***#9 - 02/15/2017 09:36 PM - Tom Clegg***- Assigned To set to Tom Clegg***#10 - 02/15/2017 10:00 PM - Tom Clegg***- Description updated***#11 - 02/20/2017 08:46 PM - Tom Clegg**[4f0e07d462b7860bb10686c27fac16970220377f](#) with updated test case.**#12 - 02/21/2017 07:58 PM - Radhika Chippada**

- I think moving "select\_readable\_pdh" to the line above the declaration of "candidates" at line 85 would help improve readability since the rest of the clauses are building on "candidates"
- [http://doc.arvados.org/api/methods/container\\_requests.html#container\\_reuse](http://doc.arvados.org/api/methods/container_requests.html#container_reuse) needs to be updated regarding the reuse of a "completed" container. Currently, it still stays there won't be any reuse when multiple completed jobs are found
- We talked about potentially removing output or log on the oldest completed container, if it is not desirable that it be reused. However, it appears that the output or log on a container in completed state can no longer be updated. So how can this be done? Do you mean that either one of these be removed from keep? Do we need to add a blurb about this also in the above documentation?

**#13 - 02/21/2017 08:36 PM - Tom Clegg**

Radhika Chippada wrote:

- I think moving "select\_readable\_pdh" to the line above the declaration of "candidates" at line 85 would help improve readability since the rest of the clauses are building on "candidates"

Indeed, rearranged this.

- [http://doc.arvados.org/api/methods/container\\_requests.html#container\\_reuse](http://doc.arvados.org/api/methods/container_requests.html#container_reuse) needs to be updated regarding the reuse of a "completed" container. Currently, it still stays there won't be any reuse when multiple completed jobs are found

Updated, thanks.

- We talked about potentially removing output or log on the oldest completed container, if it is not desirable that it be reused. However, it appears that the output or log on a container in completed state can no longer be updated. So how can this be done? Do you mean that either one of these be removed from keep? Do we need to add a blurb about this also in the above documentation?

Yes, trashing the output or log collection would accomplish this. I added to the docs "...whose log and output collection are still available". Documenting the "poking re-use in the eye" procedure seems worthwhile too but it's more of a workflow trick than API documentation -- e.g., you could make use of that information even if you only use Workbench and don't know what an API is. Wiki?

→ [802af81e13dd11a7f2d9796a2ada8faf3b722477](https://github.com/arvados/arvados/commit/802af81e13dd11a7f2d9796a2ada8faf3b722477)

#### #14 - 02/21/2017 08:57 PM - Radhika Chippada

Yes, trashing the output or log collection would accomplish this ... "poking re-use in the eye" procedure seems worthwhile too but it's more of a workflow trick than API documentation -- e.g., you could make use of that information even if you only use Workbench and don't know what an API is. Wiki?

I'd imagine someone would ask how to do this in no time. So, please add a note wherever you think appropriate. Thanks.

LGTM

#### #15 - 02/22/2017 06:25 PM - Tom Clegg

- Status changed from *In Progress* to *Resolved*

- % Done changed from 50 to 100

Applied in changeset arvados|commit:0c529ed05805507b4d2c903b9587e9b61cec5ee6.