

Arvados - Feature #8018

[Crunch2] Identify container failure and retry

12/15/2015 08:46 PM - Peter Amstutz

Status: Resolved	Start date: 09/23/2016
Priority: Normal	Due date:
Assigned To: Peter Amstutz	% Done: 100%
Category:	Estimated time: 0.00 hour
Target version: 2016-10-12 sprint	
Description When a container goes into cancelled state, create a new duplicate container when these conditions are met by at least one container request: <ul style="list-style-type: none">request is in "committed" state (not uncommitted or finalized)request has priority > 0request container_count < container_count_max Add a new column, container_count. Each time a new container is assigned to a committed container request, increment container_count.	
Subtasks:	
Task # 10062: Review 8018-container-retry	Resolved
Task # 10064: Retry on container cancelled	Resolved
Task # 10063: Determine how to count retry attempts	Resolved
Related issues:	
Related to Arvados - Feature #14706: [Crunch2] Retain references + permission...	Resolved

Associated revisions

Revision 47e42f11 - 10/07/2016 06:32 PM - Peter Amstutz

Merge branch '8018-container-retry' closes #8018

History

#1 - 02/14/2016 11:58 PM - Peter Amstutz

- Description updated

#2 - 06/13/2016 08:56 PM - Tom Clegg

So far, aside from the hint "retry" in the subject line, this reads like a solution in search of a problem. I'm assuming the relevant features we're hoping to support here are:

- Automatically retry failed containers when appropriate (so users don't have to push "try again" every time there's a transient infrastructure problem)
- Re-use existing containers when appropriate

(In both cases, the definition of "appropriate" is one thing we need to figure out.)

Among others, we should consider the use case where

- Container exits zero, state=Complete
- User inspects the output and concludes the container failed
- User wants to mark the container as "failed" in some way, so it doesn't get reused in future workflows

One way the user can address this is to fix the code that misreported "success": this would result in new docker images or git trees, and the workflow could be updated to require the new versions. This approach has some drawbacks, though:

- Until the code and workflow are fixed (which can be non-trivial), there is no notation anywhere that the output is bogus
- Updating the workflow to require a new version invalidates *all* previous containers/outputs, not just the known-bad one. This can cause an unacceptably huge amount of unnecessary computation.

#3 - 09/14/2016 07:22 PM - Peter Amstutz

- Target version set to 2016-09-28 sprint

#4 - 09/14/2016 07:35 PM - Peter Amstutz

- Story points set to 1.0

#5 - 09/14/2016 08:01 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#6 - 09/14/2016 08:03 PM - Peter Amstutz

- Description updated

#7 - 09/14/2016 08:23 PM - Tom Clegg

Suggest adding a "number of attempts already made" attribute to a container request, incrementing it atomically when retrying, and comparing it to count_max. (Using "remaining tries" is troublesome because it's hard for clients to update that atomically.)

We could expose this in API responses but prohibit clients from updating it.

#8 - 09/14/2016 08:40 PM - Peter Amstutz

Tom Clegg wrote:

Suggest adding a "number of attempts already made" attribute to a container request, incrementing it atomically when retrying, and comparing it to count_max. (Using "remaining tries" is troublesome because it's hard for clients to update that atomically.)

We could expose this in API responses but prohibit clients from updating it.

This sounds good, will update description.

#9 - 09/14/2016 08:44 PM - Peter Amstutz

- Description updated

#10 - 09/14/2016 08:44 PM - Peter Amstutz

- Description updated

#11 - 09/28/2016 05:38 PM - Peter Amstutz

- Status changed from New to In Progress

#12 - 09/28/2016 06:39 PM - Radhika Chippada

- Target version changed from 2016-09-28 sprint to 2016-10-12 sprint

#13 - 10/05/2016 12:50 AM - Radhika Chippada

- Did you forget to commit the db migration script to add the container_count?
- When retryable_requests exist in container.handle_completed, should we be using find_reusable and then create if none found instead?
- In the container_request_test, I think you can assert that the cr.container_uuid is the same as that after the first cancel related update.
- Of course, I would need to re-review after you push the migration script

#14 - 10/06/2016 04:54 PM - Peter Amstutz

Radhika Chippada wrote:

- Did you forget to commit the db migration script to add the container_count?

Oops. Committed.

- When retryable_requests exist in container.handle_completed, should we be using find_reusable and then create if none found instead?

We could do that, although because of job reuse, I don't know how likely it is that there will be another active container with exactly the same attributes, because a container request for that container would have been associated with the container that was just cancelled (which is being retried). If/when we adjust the reuse behavior (for impure containers, or just an explicit "don't reuse" flag on container request) it might not play well with the idea of reusing containers on retry.

So this is what I coded this up first:

```
reusable = Container.find_reusable c_attrs
c = if not reusable.nil?
  reusable
else
  Container.create! c_attrs
end
```

But I think I just talked myself out of it, so I'm going to put it back to always creating a new container.

```
c = Container.create! c_attrs
```

- In the `container_request_test`, I think you can assert that the `cr.container_uuid` is the same as that after the first cancel related update.

I am a little confused by what you are suggesting, but I added some additional assertions checking if `cr.container_uuid` changed or not.

- Of course, I would need to re-review after you push the migration script

Now at [8317b6e](#)

#15 - 10/07/2016 06:10 PM - Radhika Chippada

LGTM

#16 - 10/07/2016 06:35 PM - Peter Amstutz

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

- % Done changed from 67 to 100

Applied in changeset `arvados|commit:47e42f1129363c2565e69c36ff26ce9c42731fb8`.

#18 - 02/19/2019 09:13 PM - Tom Clegg

- Related to Feature #14706: *[Crunch2] Retain references + permissions to earlier containers when retrying a container request added*