Arvados - Story #9407

[Workbench] Implement container log viewer (live + saved) using work unit model

06/14/2016 06:52 PM - Brett Smith

Status:	Resolved	**Start date:**	06/27/2016
Priority:	Normal	**Due date:**	
Assigned To:	Radhika Chippada	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2016-07-06 sprint		

Description

Primary goal: Workbench only has a single log viewer, which supports all work units.

This log viewer is primarily based on the current "live job log" view. It looks more like a terminal. It can get live updates. It shows as much of the log as can reasonably fit in RAM, from most to least recent.

We want filtering capabilities, but those will be added in a separate follow-up story.

Subtasks:

Task # 9470: Review branch 9407-container-log-viewer	**Resolved**

Related issues:

Related to Arvados - Story #9514: [API] [Workbench] Delete old container log ...	**Resolved**	**06/29/2016**

Associated revisions

Revision afabb6c3 - 06/29/2016 06:03 PM - Radhika Chippada

closes #9407
Merge branch '9407-container-log-viewer'

History

#1 - 06/14/2016 06:55 PM - Brett Smith

- Description updated

- Assigned To set to Radhika Chippada

Assigned to Radhika for grooming.

#2 - 06/21/2016 02:45 PM - Radhika Chippada

Our most pressing need is to present a container's log to the user. Also, a container has only one log (similar to a job) and according to Peter it does not have a stats (and hence no graph to be displayed). This being the case, the following initial implementation of a work unit log might meet our immediate goals.

Part 1:

- If the work unit is complete, fetch the log collection and display (similar to how a job does currently)
- If the work unit is running, get the object logs from log table and display them with log event listening (similar to how a pipeline instance does currently)
- Hook this up to container#Log display

Part 2:

- The above can then easily be expanded to display a pipeline instance log
- A job log can be a superset of the above basic implementation + the missing graph display ?

#3 - 06/21/2016 07:25 PM - Tom Clegg

Might be good to sort out how we want to handle child {jobs/containers}.

Currently, the pipeline instance live log viewer shows log entries from children: without them, the live log view would be nearly pointless. Logs for completed jobs are offered as links. Meanwhile, the job log viewer doesn't even recognize the possibility of child logs: this is inconvenient when you're wondering why a job is just sitting there doing nothing, but on the bright side it does prevent the current/parent job's own logs from getting lost in a sea of child logs.

Perhaps a reasonable starting point would be

- Show live/saved logs from the current work unit, just like we do with jobs
- If there are any children, offer link to each child's log tab

#4 - 06/22/2016 07:23 PM - Radhika Chippada

- Target version set to 2016-07-06 sprint

- Story points set to 1.0

#5 - 06/22/2016 08:20 PM - Tom Clegg

- Subject changed from [Workbench] Unify log views via the work unit interface to [Workbench] Implement container log viewer (live + saved) using work unit model

#6 - 06/23/2016 01:56 PM - Radhika Chippada

- Status changed from New to In Progress

#7 - 06/23/2016 10:26 PM - Radhika Chippada

> shows as much of the log as can reasonably fit in RAM, from most to least recent

Do we have a mechanism to fetch the last N bytes of a log collection?

#8 - 06/24/2016 01:52 PM - Peter Amstutz

When the container is running, we want the live log view. Containers have multiple event types: "stdout", "stderr", "arv-mount", "crunch-run".

When the container is complete, we want links to each of the files in the container's log collection (this will contain multiple files, usually stdout.txt, stderr.txt, crunch-run.txt, arv-mount.txt)

#9 - 06/24/2016 09:19 PM - Radhika Chippada

Branch 9407-container-log-viewer at c9edd857

Implements log view for containers and container_requests as specified in note-8.

#10 - 06/27/2016 05:29 PM - Tom Clegg

Functional

(from description)

> This log viewer is primarily based on the current "live job log" view. It looks more like a terminal.

I read note-8 to mean that we *also* want links to the individual log files when applicable (i.e., non-live mode). But the current branch seems to show the generic "list of files in the collection" view *instead of* a log viewer.

My understanding (and hope) is that we're working towards making the live and finished log views as similar as possible. E.g., "download stderr as a file before container finishes", when we get around to implementing it, should probably end up in the same place as the "download stderr as a file after container finishes" button. From the user's perspective the only meaningful difference between incomplete and complete/saved logs is really just a "finished?" flag, so it seems like the UI should reflect that, rather than emphasizing the differences in how the two cases are implemented.

Code

I think the arv-log-event-handler-append-logs function in source:apps/workbench/app/assets/javascripts/pipeline_instances.js should be updated to recognize "arv-mount" and "crunch-run" -- or perhaps even better to ignore event_type, and display any log that has something in eventData.properties.text, so we don't have to keep the lists of magic words synchronized.

Can we use assert_select to match the anchor & div elements instead of matching the HTML strings literally?

Instead of object = @object unless object can we update the other use of the show_files partial to pass an object explicitly? (There seems to be only one other place where we use this partial.) Perhaps this is a moot point if we link to the log collection instead of using the "list files" partial.

In source:apps/workbench/app/views/work_unit/_show_log, we use the work unit's "log" method (shouldn't this be "log_collection"?), look up the collection, and then pass that back to render_log. How about changing the render_log method so it looks up the log collection by itself instead of asking the caller to pass it back in? It might also end up a bit nicer if render_log returns the actual arguments to pass to render, so the view could do this:

```
def render_log
  ...
  collection = Collection.find(log_collection)
  return {partial: 'collections/show_files', locals: {object: collection, no_checkboxes: true}
end

# in view:
render(wu.render_log)
```

#11 - 06/27/2016 07:53 PM - Radhika Chippada

> (from description) This log viewer is primarily based on the current "live job log" view. It looks more like a terminal... I read note-8 to mean that we also want links to the individual log files when applicable (i.e., non-live mode). But the current branch seems to show the generic "list of files in the collection" view instead of a log viewer.

Peter explicitly wanted the files to be shown in a container log viewer rather than show a link to the log collection (similar to pipeline_instances) and clicking which takes the user to the log to avoid two clicks. If there is concern about container log display being different than that of a pipeline instance, I can change this to display a link to log collection instead. In case of a running container, it does look like a terminal for a container object as well.

> My understanding (and hope) is that we're working towards making the live and finished log views as similar as possible. E.g., "download stderr as a file before container finishes", when we get around to implementing it, should probably end up in the same place as the "download stderr as a file after container finishes" button. From the user's perspective the only meaningful difference between incomplete and complete/saved logs is really just a "finished?" flag, so it seems like the UI should reflect that, rather than emphasizing the differences in how the two cases are implemented.

In case of a container, at least from what I understood, we have no other log collections while it is still running (where as a pipeline instance has log collections for the complete job children and a running log for any running jobs). Since I was told to focus on container log display only, for now, I only implemented the case where it is either running or has a log collection (completed). We currently do not have a case where we get the log_collections of all children (as is the case for a pipeline instance), which I can add if we want it at this time.

> I think the arv-log-event-handler-append-logs function in source:apps/workbench/app/assets/javascripts/pipeline_instances.js should be updated to recognize "arv-mount" and "crunch-run" -- or perhaps even better to ignore event_type, and display any log that has something in eventData.properties.text, so we don't have to keep the lists of magic words synchronized.

Updated

> Can we use assert_select to match the anchor & div elements instead of matching the HTML strings literally?

Updated

> Instead of object = @object unless object can we update the other use of the show_files partial to pass an object explicitly? (There seems to be only one other place where we use this partial.) Perhaps this is a moot point if we link to the log collection instead of using the "list files" partial.

Collection#Show has "files" tab. I do not think I can avoid having @object in this. Hence, I left this as is.

> In source:apps/workbench/app/views/work_unit/_show_log, we use the work unit's "log" method (shouldn't this be "log_collection"?), look up the collection, and then pass that back to render_log. How about changing the render_log method so it looks up the log collection by itself instead of asking the caller to pass it back in? It might also end up a bit nicer if render_log returns the actual arguments to pass to render, so the view could do this:

Improved the render_log method. (I am including the scenario where a log_collection exists but not readable by the user)

#12 - 06/29/2016 02:04 PM - Tom Clegg

Radhika Chippada wrote:

> > (from description) This log viewer is primarily based on the current "live job log" view. It looks more like a terminal... I read note-8 to mean that we also want links to the individual log files when applicable (i.e., non-live mode). But the current branch seems to show the generic "list of files in the collection" view instead of a log viewer.
>
> Peter explicitly wanted the files to be shown in a container log viewer rather than show a link to the log collection (similar to pipeline_instances) and clicking which takes the user to the log to avoid two clicks. If there is concern about container log display being different than that of a pipeline instance, I can change this to display a link to log collection instead. In case of a running container, it does look like a terminal for a container object as well.

I'm on board with minimizing clicks, and reusing the generic collection-browsing view seems like a good idea. My concern is just that for some reason we aren't *also* showing the actual log content. IOW: While my container is running, I watch its log messages scroll by in a little terminal window -- but if I hit Refresh, and the container has finished, the tab loses its most useful feature: the terminal window with the actual log messages disappears, and instead I have to click "preview" or "download" to see logs. Shouldn't the terminal window feature aim for "show logs if possible" + "download logs if possible", rather than "show logs if downloading is impossible"?

> My understanding (and hope) is that we're working towards making the live and finished log views as similar as possible. E.g., "download stderr as a file before container finishes", when we get around to implementing it, should probably end up in the same place as the "download stderr as a file after container finishes" button. From the user's perspective the only meaningful difference between incomplete and complete/saved logs is really just a "finished?" flag, so it seems like the UI should reflect that, rather than emphasizing the differences in how the two cases are implemented.

> In case of a container, at least from what I understood, we have no other log collections while it is still running (where as a pipeline instance has log collections for the complete job children and a running log for any running jobs). Since I was told to focus on container log display only, for now, I only implemented the case where it is either running or has a log collection (completed). We currently do not have a case where we get the log_collections of all children (as is the case for a pipeline instance), which I can add if we want it at this time.

I didn't mean to ask for a "download live log" feature right now. I was just using it to make the point that the implementation details should serve the UX, rather than the other way around.

> Instead of object = @object unless object can we update the other use of the show_files partial to pass an object explicitly? (There seems to be only one other place where we use this partial.) Perhaps this is a moot point if we link to the log collection instead of using the "list files" partial.

> Collection#Show has "files" tab. I do not think I can avoid having @object in this. Hence, I left this as is.

Ah, right, the generic tab display code (render_index in source:apps/workbench/app/controllers/application_controller.rb#L179) calls the same partial. I think we could pass {object: @object} in locals here too (in general it seems better to use locals instead of controller instance variables for views), but if this is a can of worms let's leave it alone.

#13 - 06/29/2016 02:56 PM - Radhika Chippada

> My concern is just that for some reason we aren't also showing the actual log content ...

Updated to show the log terminal window in the case of a finished container also.

> I didn't mean to ask for a "download live log" feature right now. I was just using it to make the point that the implementation details should serve the UX, rather than the other way around.

As we talked about it, no action about it.

> Ah, right, the generic tab display code (render_index in source:apps/workbench/app/controllers/application_controller.rb#L179) calls the same partial. I think we could pass {object: @object} in locals here too (in general it seems better to use locals instead of controller instance variables for views), but if this is a can of worms let's leave it alone.

Yes, I think it could be a can of worms. Leaving it alone. Thanks.

#14 - 06/29/2016 04:28 PM - Tom Clegg

LGTM @ 266a9bd, thanks.

#15 - 06/29/2016 06:25 PM - Radhika Chippada

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset arvados|commit:afabb6c30c449d6139aec344d0912fc2645e2e89.