Arvados - Feature #3899

[Crunch] crunch-job and Workbench use the new job state attribute (see #3898) instead of paying attention to (and updating) running and success flags.

09/17/2014 02:58 PM - Tom Clegg

Status:	Resolved		**Start date:**	09/24/2014
Priority:	Normal		**Due date:**	
Assigned To:	Peter Amstutz		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:	2014-10-08 sprint			

Description

Expected changes:

- Crunch-job reads/writes the new state attribute instead of the old success/running flags.
- Workbench uses the state attribute provided by API, instead of deriving its own state string by comparing running/success/cancelled_at etc.
- ditto arv-run-pipeline-instance

Subtasks:

Task # 3962: Update crunch-dispatch	**Resolved**
Task # 3963: Update arv-run-pipeline-instance	**Resolved**
Task # 3982: Review 3988-crunch-use-job-state	**Resolved**
Task # 3921: Update crunch-job	**Resolved**

Associated revisions

Revision 1b189a09 - 09/26/2014 02:31 PM - Peter Amstutz

Merge branch '3899-crunch-use-job-state' closes #3899

Revision e6fbedf9 - 10/01/2014 02:00 AM - Peter Amstutz

3899: Fix for counting complete/failed components refs #3899

History

#1 - 09/17/2014 04:18 PM - Peter Amstutz

- Target version changed from Arvados Future Sprints to 2014-10-08 sprint

#2 - 09/17/2014 04:18 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

- Story points changed from 1.0 to 2.0

#3 - 09/23/2014 07:49 PM - Peter Amstutz

Design proposal to implement this story in a way that also fixes #3859

- Changing state from "queued" to "running" also sets is_locked_by_uuid and started_at. Fails if the starting state is not "queued".
- Change is_locked_by_uuid to is_locked_by_token which is the API token that was issued to job instead of the user uuid. Restrict is_locked_by_token column from everybody except admins and crunch-dispatch.
- Restrict further changes to the job record to that API token.
- Set finished_at when state changes from "Running" to one of Complete, Cancelled, Failed

#4 - 09/23/2014 08:07 PM - Peter Amstutz

"Consider what happens when running a job in local mode"

#5 - 09/23/2014 08:37 PM - Tom Clegg

- Description updated

#6 - 09/25/2014 04:14 PM - Tom Clegg

Notes @ 9daebff...

In arv-run-pipeline-instance the map/reduce bits deserve to be less verbose. Suggest something like this:

- ███████████████████████████

```
-         if c[:job] and ["Complete", "Failed", "Cancelled"].include? c[:job][:state] then 1 else 0 end
-     }.reduce(:+) || 0
-
-     succeeded = @components.map { |cname, c|
-       if c[:job] and ["Complete"].include? c[:job][:state] then 1 else 0 end
-     }.reduce(:+) || 0
-
-     failed = @components.map { |cname, c|
-       if c[:job] and ["Failed", "Cancelled"].include? c[:job][:state] then 1 else 0  end
-     }.reduce(:+) || 0

+     c_in_state = @components.values.group_by { |c|
+       c[:job] and c[:job][:state]
+     }
+     succeeded = c_in_state["Complete"].count
+     failed = c_in_state["Failed"].count + c_in_state["Cancelled"].count
+     ended = succeeded + failed
```

crunch-job: ooh, nice secret bugfix:

- ████████████████████████████████

```
+exit ($Job->{'state'} != 'Complete' ? 1 : 0);
```

crunch-job: This error message could contain misleading information if the state changes to something other than Cancelled. I think it's worth separating the expected case ("Job cancelled at ... by ...") from the unexpected case ("Job state unexpectedly changed to X. Who could work this way? I give up."), which would also avoid the misleading message.

- ██████████████████████████████

```
-         Log (undef, "Job cancelled at " . $Job->{cancelled_at} .
+     if ($Job->{'state'} ne "Running") {
+         Log (undef, "Job state changed to " . $Job->{'state'} . " at " . $Job->{cancelled_at} .
+             " by user " . $Job->{cancelled_by_user_uuid});
```

crunch-job: I think this should test state==Cancelled instead of testing cancelled_at, and therefore shouldn't bother setting state to Cancelled:

- ████████████████████████

```
+     $Job->update_attributes('state' => 'Cancelled',
+                             'finished_at' => scalar gmtime);
```

Todo:

- workbench pipeline_instances_helper.rb should use new state attr
- workbench Job.state class method should go away (afaict only app/views/application/_job_status_label.html.erb uses this, presumably it should now use "j.state" instead of "Job.state j")

#7 - 09/25/2014 07:04 PM - Peter Amstutz

Tom Clegg wrote:

> Notes @ [9daebff](#)...
>
> In arv-run-pipeline-instance the map/reduce bits deserve to be less verbose. Suggest something like this:
>
> - [...]

Nice! Fixed.

> crunch-job: This error message could contain misleading information if the state changes to something other than Cancelled. I think it's worth separating the expected case ("Job cancelled at ... by ...") from the unexpected case ("Job state unexpectedly changed to X. Who could work this way? I give up."), which would also avoid the misleading message.
>
> - [...]

Fixed.

> crunch-job: I think this should test state==Cancelled instead of testing cancelled_at, and therefore shouldn't bother setting state to Cancelled:

- [...]

Fixed.

   Todo:

   - workbench pipeline_instances_helper.rb should use new state attr

Fixed.

   - workbench Job.state class method should go away (afaict only app/views/application/_job_status_label.html.erb uses this, presumably it should now use "j.state" instead of "Job.state j")

If we get rid of Job.state it breaks the one specific case where we're looking at an old pipeline instance record and for some reason the actual job object isn't available so we're looking at the unmigrated copy in 'components'. Unless you feel strongly about it, I would prefer to leave it since it's more effort to change it (also Job::state is used in several other partials).

#8 - 09/25/2014 07:37 PM - Tom Clegg

Now at [2861857](#)

Peter Amstutz wrote:

   > workbench Job.state class method should go away (afaict only app/views/application/_job_status_label.html.erb uses this, presumably it should now use "j.state" instead of "Job.state j")
   >
   > If we get rid of Job.state it breaks the one specific case where we're looking at an old pipeline instance record and for some reason the actual job object isn't available so we're looking at the unmigrated copy in 'components'. Unless you feel strongly about it, I would prefer to leave it since it's more effort to change it (also Job::state is used in several other partials).

Ah, I see now. Perhaps adding a state field when needed in the pipeline helper's "make a fake job object" might be a more appropriate way to do this, though? That way Job::state j could get replaced with j[:state] everywhere, instead of having that mystery scattered throughout and expected of all future code.

Everything else LGTM, thanks!

#9 - 09/26/2014 12:40 AM - Peter Amstutz

- Status changed from New to In Progress

#10 - 09/26/2014 01:31 AM - Peter Amstutz

Tom Clegg wrote:

   > Ah, I see now. Perhaps adding a state field when needed in the pipeline helper's "make a fake job object" might be a more appropriate way to do this, though? That way Job::state j could get replaced with j[:state] everywhere, instead of having that mystery scattered throughout and expected of all future code.

Fixed, but I can't test it because of other bugs rendering the 'job' portion of pipeline components. I think Tim is working on this so I don't want to duplicate his work.

#11 - 09/26/2014 02:09 PM - Tom Clegg

Missing newline at EOF @services/api/test/fixtures/jobs.yml.

In apps/workbench/app/helpers/pipeline_instances_helper.rb I suspect we should leave pj[:job][:state] alone if it's already there.

Thanks

#12 - 09/26/2014 02:20 PM - Peter Amstutz

- Status changed from In Progress to New

Tom Clegg wrote:

   > Missing newline at EOF @services/api/test/fixtures/jobs.yml.

Fixed.

In apps/workbench/app/helpers/pipeline_instances_helper.rb I suspect we should leave pj[:job][:state] alone if it's already there.

Fixed.

#13 - 09/26/2014 02:35 PM - Anonymous

- Status changed from New to Resolved

- % Done changed from 75 to 100

Applied in changeset arvados|commit:1b189a0961ba757caf6160285b59daa26c7cdcae.

#13 - 09/26/2014 02:35 PM - Anonymous

- Status changed from New to Resolved