Status: Resolved  
Start date: 05/14/2019

Priority: Normal
Due date: % Done: 100%

Assigned To: Lucas Di Pentima
Estimated time: 0.00 hour

Category: Workbench

Target version: 2019-05-22 Sprint

Description
Rails 4.2 is about to be EOLed when 6.0 is released.

Subtasks:
Task # 15032: Review 14988-wb-rails5-upgrade
Resolved

Related issues:
Blocks Arvados - Story #14946: Update to Ruby 2.5 - 2.3 is going EOL
Resolved 05/31/2019

Revision fd5d1979 - 05/17/2019 05:31 PM - Lucas Di Pentima
Merge branch '14988-wb-rails5-upgrade'
Closes #14988

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

History
#1 - 03/18/2019 05:43 PM - Lucas Di Pentima
- Blocks Story #14946: Update to Ruby 2.5 - 2.3 is going EOL added

#2 - 03/27/2019 03:14 PM - Lucas Di Pentima
- Target version changed from 2019-03-27 Sprint to 2019-04-10 Sprint

#3 - 03/27/2019 03:42 PM - Tom Morris
- Story points set to 3.0

#4 - 04/09/2019 01:36 PM - Lucas Di Pentima
- Status changed from New to In Progress

#5 - 04/10/2019 01:06 PM - Lucas Di Pentima
- Target version changed from 2019-04-10 Sprint to 2019-04-24 Sprint

#6 - 04/24/2019 03:23 PM - Lucas Di Pentima
- Target version changed from 2019-04-24 Sprint to 2019-05-08 Sprint

#7 - 04/24/2019 05:09 PM - Lucas Di Pentima
- Story points changed from 3.0 to 2.0

#8 - 04/25/2019 02:15 PM - Lucas Di Pentima

Status update
Updates at 0ebe4a09a3d12f1415d2c119e716a2788f889ec - branch 14988-wb-rails5-upgrade
- I've been eliminating the deprecation warnings after doing the rake app:update call.
- Forked a couple of gems because they aren't being maintained:
  - activerecord-nulldb-adapter
  - wiselinks: This one is not yet clear that it works, I was able to make it stop complaining about deprecated functions but it's from rails 3.x and it hasn't being updated since then. I've tried to submit a PR to the original maintainer but their travis pipeline fail because it's set up to test against ruby 1.9 & 2.0, and some of the updated dependencies aren't compatible with those ruby environments.
I'm now trying to make the unit test pass, but our fake ActiveRecord::Column loading is not working properly, I think in part it's because the ActiveRecord::Column API changed again (we had issues when upgrading from 4.1 to 4.2) and I'm not getting much documentation about how to work with it, so what I'm trying is the new ActiveRecord::Attributes API (https://api.rubyonrails.org/classes/ActiveRecord/Attributes/ClassMethods.html) that seems to be the correct way to go about adding attributes not backed by a database.

#9 - 04/25/2019 02:34 PM - Tom Clegg
It's possible that we don't even benefit from wiselinks and removing it would have no effect. Worth trying, if it causes any further trouble.

As for the fake-activerecord stuff, all I can think to offer is that IIRC it has always been a hack, written in "whatever seems to work" style, so it's not too surprising to hear Rails upgrades keep breaking it.

#10 - 05/08/2019 04:45 PM - Lucas Di Pentima
- Target version changed from 2019-05-08 Sprint to 2019-05-22 Sprint
- Story points changed from 2.0 to 0.5

#11 - 05/13/2019 04:25 PM - Lucas Di Pentima
Updates at 28ca19d77 - branch 14988-wb-rails5-upgrade
Test run: https://ci.curoverse.com/job/developer-run-tests/1234/

All tests finally passing. Pending items:
- Update bootstrap-sass gem to version 3.4.1 to address CVE-2019-8331
- Fix test runner call to be able to run individual test files.

#12 - 05/14/2019 03:09 PM - Lucas Di Pentima
Updated bootstrap-sass at 2e110679f

#13 - 05/14/2019 07:01 PM - Tom Clegg
I'm not sure whether anything will care, but this (in arvados_base.rb) seems like it would accidentally match new records:

```ruby
def destroyed?
  !(etag || uuid)
end
```

This (arvados_base.rb) seems like it silently returns nil for nonexistent columns, and foo['bar'] where bar happens to be a method that raises. It looks like this is replacing Rails.logger.debug "BUG: access non-loaded attribute #{k}" -- did you happen to find out what's relying on this?

```ruby
def [](attr_name)
  begin
    send(attr_name)
  rescue
    nil
  end
end
```

#14 - 05/17/2019 03:13 PM - Lucas Di Pentima
Tom Clegg wrote:

I'm not sure whether anything will care, but this (in arvados_base.rb) seems like it would accidentally match new records:

Fixed, thanks!

This (arvados_base.rb) seems like it silently returns nil for nonexistent columns, and foo['bar'] where bar happens to be a method that raises. It looks like this is replacing Rails.logger.debug "BUG: access non-loaded attribute #{k}" -- did you happen to find out what's relying on this?

Restored the debug message. One of the parts relying on this is apps/workbench/app/views/containers/_show_log.html.erb, not sure why it is calling object[:name] but since this is a rails migration story, I didn't want to go into that rabbit hole :)

Updates at 17ede5d13
Test run: https://ci.curoverse.com/job/developer-run-tests/1241/

#15 - 05/17/2019 05:21 PM - Tom Clegg
Thanks, LGTM
#16 - 05/17/2019 05:33 PM - Lucas Di Pentima
- Status changed from In Progress to Resolved
- % Done changed from 0 to 100

Applied in changeset arvados|fd5d197916b117386fbe872790ff6493326c4ea2.

#17 - 05/21/2019 10:27 PM - Tom Morris
- Release set to 15