

## Arvados - Feature #4561

### [SDKs] Refactor run-command so it can be used as an SDK by scripts in a git tree

11/17/2014 09:10 PM - Tom Clegg

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>		<b>% Done:</b>	0%
<b>Category:</b>	SDKs	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Arvados Future Sprints		
<b>Description</b>			
Currently you have two main options for running a job:			
<ol style="list-style-type: none"><li>1. Put the program you need in a docker image. Use run-command from the arvados tree to wrap it as a crunch job.</li><li>2. Write a native crunch script in your git tree.</li></ol>			
The first option forces you to save a new docker image in order to run a new version of your program. Unacceptable!			
There are several features of run-command that make it convenient and attractive to beginners. However, it forces you to use a totally different approach to developing and running scripts, an approach which prevents you from doing some important things:			
<ul style="list-style-type: none"><li>• Keeping your code in revision control,</li><li>• Using the same code in more than one pipeline template,</li><li>• Writing jobs in the programming language of your choice.</li></ul>			
There is no migration path from a simple run-command job to a non-trivial program, so the developer is forced to choose: live with run-command's custom JSON-based programming language, or abandon the existing pipelines and all of run-command's advantages, and rewrite everything in a normal language like Python.			
This can be addressed by refactoring run-command as a set of utilities and features, rather than a programming language that can only be used inside crunch jobs.			
<ul style="list-style-type: none"><li>• The "run-command language" interpreter should be runnable like other interpreters (<code>#!/whatever-the-language-is-to-be-called</code>).</li><li>• Convenience features like "store output dir contents in Keep and set success=true at end of task" should be ported to other SDKs too (most obviously bash), so authors can migrate from the JSON language to a normal language.</li><li>• It should be possible to provide a crunch script in <i>any</i> language by copying the script itself into the job record. (This makes it possible to run jobs without touching git.)<ul style="list-style-type: none"><li>◦ The whole point of the run-command language is to already be JSON-encoded, which means it should be provided in a serialized attribute like <code>script_parameters</code>. (Other languages are just text, but that can also fit in a serialized field.)</li><li>◦ The name of the "script" attribute already suggests that you can put a script in it. This could change type (from <code>varchar(255)</code> to text) and become a serialized field capable of containing a string or a hash. This means we can't rely on "script" to be a short name suitable for displaying in UI (a problem we already have with run-command jobs: the program has no name, so we display the name of the language instead).</li><li>◦ We could support passing the name of an interpreter in "script" (e.g., "run-command" or "python") and passing the program itself in <code>script_parameters[stdin]</code>. We would have to treat string and hash/array cases differently: if <code>stdin</code> is a string, pass the string value, but if <code>stdin</code> is a hash or array, pass its JSON encoding.</li></ul></li><li>• It should be possible to move your run-command program into a git tree and run it from there.<ul style="list-style-type: none"><li>◦ Currently, this can be done awkwardly by copying some version of run-command into your own git tree.</li><li>◦ With <a href="#">#4027</a>, we can make run-command's features available through the SDK.</li><li>◦ Then we just need <code>#!/whatever-the-language-is-called</code> to work, or some other way to invoke run-command from the installed SDK, rather than requiring it to be in <code>\$CRUNCH_SRC/crunch_scripts/</code>.</li></ul></li></ul>			
<b>Related issues:</b>			
Related to Arvados - Bug #4562: [Documentation] Wiki page: explain appropriat...		<b>Resolved</b>	<b>01/16/2015</b>
Related to Arvados - Story #3820: [Crunch] Jobs need code from multiple git r...		<b>Closed</b>	<b>09/05/2014</b>
Related to Arvados - Story #3603: [Crunch] Design good Crunch task API, inclu...		<b>Closed</b>	<b>08/27/2014</b>

#### History

#1 - 11/17/2014 09:18 PM - Tom Clegg

- Description updated

- Category set to Crunch

**#2 - 01/20/2015 07:56 PM - Tom Clegg**

- Subject changed from *[Crunch] Support using run-command to wrap a script in a separate git tree.* to *[SDKs] Refactor run-command so it can be used as an SDK by scripts in a git tree*
- Description updated
- Category changed from *Crunch* to *SDKs*
- Target version set to *Arvados Future Sprints*
- Story points changed from *1.0* to *2.0*

**#3 - 02/04/2015 03:47 PM - Tom Clegg**

- Description updated

**#4 - 02/04/2015 03:55 PM - Tom Clegg**

- Description updated

**#5 - 02/04/2015 04:01 PM - Peter Amstutz**

(comments from IRC)

Now that we have the "deploy SDK into container" feature, moving most of run-command's functionality into the SDK is a good idea. The main reason I didn't do that was because the deployment cycle for the SDK was much less convenient than for a crunch script. In the long term I'd prefer to deprecate run-command in favor of CWL (and send run-command to the scrapyard for spare parts).