

Arvados - Story #8653

[CWL SDK] cwl-runner can run a pipeline instance that was submitted by Workbench using a template from "cwl-runner --create-template"

03/07/2016 04:45 PM - Brett Smith

Status:	Resolved	Start date:	03/07/2016
Priority:	Normal	Due date:	
Assigned To:	Peter Amstutz	% Done:	100%
Category:	SDKs	Estimated time:	0.00 hour
Target version:	2016-05-25 sprint		
Description			
Add a tool to the CWL SDK called ??? that creates an Arvados pipeline template to run a CWL workflow.			
<ul style="list-style-type: none">• It has a switch or optional position parameter to update an existing pipeline template identified by its UUID.			
The tool must:			
<ul style="list-style-type: none">• Where static files are identified in the workflow (including as defaults), upload them to collections.• Write a pipeline template to run the workflow with cwl-runner. Specific handling of inputs:<ul style="list-style-type: none">◦ Where static files were inputs, change those inputs to refer to the corresponding collection portable data hashes.◦ Convert CWL input "label" to template "title".◦ Convert CWL input "description" to template "description".◦ Convert CWL input "default" to template "default". In this case, set "link_name" to give the default input a nice human-readable name.◦ When the CWL input is of a type that Workbench doesn't support (i.e., the type is not documented in the pipeline template API documentation), print a warning on stderr that the template isn't runnable through Workbench. (We'll adjust this warning in the future as we improve the Workbench input chooser.)			
Subtasks:			
Task # 9024: test wl with all inputs specified			Resolved
Task # 9190: Update cwl-runner crunch script			Resolved
Task # 9025: test wl with some inputs not specified			Resolved
Task # 8986: Review 8653-cwl-to-template			Resolved
Task # 9192: Review 8653-cwl-runner-handle-files			Resolved

Associated revisions

Revision 25505fc5 - 05/02/2016 08:27 PM - Tom Clegg

Merge branch '8653-cwl-to-template'

refs #8653

Revision 7ea58af5 - 05/03/2016 01:23 PM - Peter Amstutz

Update version pin for cwltool fpm packages. refs #8653

Revision 7a430017 - 05/19/2016 08:02 PM - Peter Amstutz

Merge branch '8653-cwl-runner-handle-files' closes #8653

Revision fedd51d0 - 05/24/2016 05:12 PM - Ward Vandewege

Build distribution packages for the version of python-cwltool that python-arvados-cwl-runner now depends on.

refs #8653

Revision d3bd05e2 - 05/24/2016 07:41 PM - Ward Vandewege

Build distribution packages for the version of python-schema-salad that python-cwltool now depends on.

refs #8653

History

#1 - 03/08/2016 07:39 PM - Brett Smith

- Subject changed from [CWL] Workflow registration tool to [CWL SDK] Tool to create/update an Arvados pipeline template from a workflow
- Description updated
- Category changed from Crunch to SDKs

#2 - 03/08/2016 07:40 PM - Brett Smith

- Target version set to Arvados Future Sprints

#3 - 03/08/2016 07:43 PM - Brett Smith

- Story points set to 3.0

#4 - 03/16/2016 07:42 PM - Peter Amstutz

- Story points changed from 3.0 to 2.0

#5 - 04/13/2016 07:23 PM - Brett Smith

- Assigned To set to Tom Clegg
- Target version changed from Arvados Future Sprints to 2016-04-27 sprint

#6 - 04/13/2016 07:41 PM - Peter Amstutz

The starting point for this is `arvados_cwl.RunnerJob.run()`. This handles all the details to submit a job for running, so for this ticket you need the preparation steps (find & upload file dependencies, upload docker images) but then produce a pipeline template instead of a Job.

#7 - 04/18/2016 09:50 PM - Tom Clegg

- Status changed from New to In Progress

#8 - 04/25/2016 05:23 PM - Tom Clegg

In this story we want to encode a pipeline template's `script_parameters` such that Workbench can display values and offer choosers, which will look something like this:

1. `cwl-runner` creates pipeline template: `"input":{"dataclass":"Collection","value":"99999/foo.txt"}`
2. Workbench creates pipeline instance: `"input":{"dataclass":"Collection","value":"99999/foo.txt"}`
3. `arv-run-pipeline-instance` submits job: `"input":"99999/foo.txt"`

...but it looks like the resulting job will fail because it expects `cwl-style` parameters like `"input":{"class":"File","path":"99999/foo.txt"}`. Should this be a separate story, or should we add it to the requirements here? It seems like it has to be resolved in order for the `--create-template` feature to be useful.

`arvados-cwl-runner` uses `cwltool`'s `"main"` function, which returns an exit code (0 for success). If the `arvExecutor()` function returns `None`, that gets propagated to `main`'s return value as a failure. If `arvExecutor()` returns something else, that thing is printed on `stdout` (JSON-formatted, with no trailing newline), which isn't quite what you'd expect from a CLI tool. The existing `"arvados-cwl-runner --submit"` feature seems to deal with this conundrum by returning `None` and exiting non-zero, but this seems to make it unscriptable. I've written `"arvados-cwl-runner --create-template ..."` so it returns the UUID of the new template. If my reading is correct here, we probably update the `"--submit"` case to do something similar (and its test cases could be updated to check exit codes).

#9 - 04/25/2016 09:06 PM - Tom Clegg

8653-cwl-to-template @ [3610b23](#)

#10 - 04/27/2016 01:40 PM - Peter Amstutz

Tom Clegg wrote:

In this story we want to encode a pipeline template's `script_parameters` such that Workbench can display values and offer choosers, which will look something like this:

1. `cwl-runner` creates pipeline template: `"input":{"dataclass":"Collection","value":"99999/foo.txt"}`
2. Workbench creates pipeline instance: `"input":{"dataclass":"Collection","value":"99999/foo.txt"}`
3. `arv-run-pipeline-instance` submits job: `"input":"99999/foo.txt"`

...but it looks like the resulting job will fail because it expects `cwl-style` parameters like `"input":{"class":"File","path":"99999/foo.txt"}`. Should this be a separate story, or should we add it to the requirements here? It seems like it has to be resolved in order for the `--create-template` feature to be useful.

When we discussed this previously, I think we decided to teach the `cwl-runner` crunch script to match string parameters that looked like keep references and turn them into CWL references (I guess we didn't write down that decision). I agree this should be part of this story.

For this script_parameters input:

```
{
  "input": "99999/foo.txt"
}
```

is translated inside cwl-runner, before being passed to main():

```
{
  "input": {
    "class": "File",
    "path": "keep:99999/foo.txt"
  }
}
```

#11 - 04/27/2016 02:08 PM - Peter Amstutz

- Looking at def pipeline_component_spec(self):
 - The CWL field "label" corresponds to "title" (this is in the story description...)
 - Use `shortname(param['id'])` to get `param_id` from `param['id']`
 - It seems a little weird to copy the CWL input parameter dict and update it, instead of just filling in a new dict
 - You should fill in defaults explicitly. I suggest calling `cwltool.process.fillInDefaults(self.tool.tool["inputs"], job_params)` at the top, this will fill in the `job_params` dict with default values for any missing values. (You will need to bump the `cwltool` version dependency, that function was added in 1.0.20160425140546).
 - `spec['script_parameters']['cwl:tool'] = job_params['cwl:tool']` I think this is a tautology because `job_params = spec['script_parameters']` ?

#12 - 04/27/2016 02:27 PM - Peter Amstutz

Tom Clegg wrote:

arvados-cwl-runner uses `cwltool`'s "main" function, which returns an exit code (0 for success). If the `arvExecutor()` function returns `None`, that gets propagated to `main`'s return value as a failure. If `arvExecutor()` returns something else, that thing is printed on `stdout` (JSON-formatted, with no trailing newline), which isn't quite what you'd expect from a CLI tool. The existing "arvados-cwl-runner --submit" feature seems to deal with this conundrum by returning `None` and exiting non-zero, but this seems to make it unscriptable. I've written "arvados-cwl-runner --create-template ..." so it returns the UUID of the new template. If my reading is correct here, we probably update the "--submit" case to do something similar (and its test cases could be updated to check exit codes).

I agree, returning the UUID string is what it should do. I just tweaked `cwltool` so that if the return value is a string it writes it directly (no json encoding), so you'll need to bump the `cwltool` version pin in `setup.py` to 1.0.20160427142240. Also agree that --submit should do the same thing.

#13 - 04/27/2016 06:59 PM - Tom Clegg

Peter Amstutz wrote:

- The CWL field "label" corresponds to "title" (this is in the story description...)

Done. (I was using an example from the cwl docs which had no labels but lots of "first line works as a title" descriptions.)

- Use `shortname(param['id'])` to get `param_id` from `param['id']`

Done. I figured there must be a function for that, but somehow I didn't find it. Thanks.

- It seems a little weird to copy the CWL input parameter dict and update it, instead of just filling in a new dict

This way leaves alone whatever entries we don't know how to translate. "Leave alone" and "remove" strategies both seem defensible, though. Do you think we should switch to "remove"?

- You should fill in defaults explicitly. I suggest calling `cwltool.process.fillInDefaults(self.tool.tool["inputs"], job_params)` at the top, this will fill in the `job_params` dict with default values for any missing values. (You will need to bump the `cwltool` version dependency, that function was added in 1.0.20160425140546).

Can't quite do that: `fillInDefaults` aborts as soon as it notices a required input with no value. I think we're OK anyway, though, because `Workbench` already knows how to use "default"...

- `spec['script_parameters']['cwl:tool'] = job_params['cwl:tool']` I think this is a tautology because `job_params = spec['script_parameters']` ?

Not quite...

```
job_params = spec['script_parameters']
```

```
spec['script_parameters'] = {}
```

Added comment to help reduce future head-scratching:

```
# Most of the component spec is exactly the same as the job
# spec (script, script_version, etc.).
# spec['script_parameters'] isn't right, though. A component
# spec's script_parameters hash is a translation of
# self.tool.tool['inputs'] with defaults/overrides taken from
# the job order. So we move the job parameters out of the way
# and build a new spec['script_parameters'].
```

→ [5dc0047](#)

#14 - 04/27/2016 07:05 PM - Tom Clegg

- Target version changed from 2016-04-27 sprint to 2016-05-11 sprint

#15 - 04/27/2016 07:50 PM - Tom Clegg

- Story points changed from 2.0 to 1.0

#16 - 05/02/2016 07:24 PM - Peter Amstutz

```
$ arvados-cwl-runner --create-template 1st.cwl
/home/peter/work/scripts/venv/bin/arvados-cwl-runner 1.0.20160427185811, arvados-python-client 0.1.20160412193
510, cwltool 1.0.20160427142240
usage: 1st.cwl [-h] --message MESSAGE [job_order]
1st.cwl: error: argument --message is required
```

```
$ arvados-cwl-runner --create-template 1st.cwl empty.yml
/home/peter/work/scripts/venv/bin/arvados-cwl-runner 1.0.20160427185811, arvados-python-client 0.1.20160412193
510, cwltool 1.0.20160427142240
2016-05-02 15:21:19 arvados.arv-run[16359] INFO: Upload local files: "1st.cwl"
2016-05-02 15:21:19 arvados.arv-run[16359] INFO: Uploaded to 3d0ga-4zz18-hyszdi9bvw9gz6t
2016-05-02 15:21:19 arvados.cwl-runner[16359] INFO: Created template 3d0ga-p5p6p-i9g2dkic7do661g
3d0ga-p5p6p-i9g2dkic7do661g
```

The argument parsing in arvados-cwl-runner is inheriting from cwltool, and in this case it's doing the wrong thing. It shouldn't block the branch but I'm noting it because it does need to be tweaked. Also arvados-cwl-runner is inheriting a bunch of other command line options from cwltool that are not meaningful, which also should be cleaned up.

#17 - 05/02/2016 08:03 PM - Peter Amstutz

8653-cwl-to-template LGTM @ [5dc0047](#)

crunch_scripts/cwl-runner still needs to be updated per note #10, can't close the ticket until that's done.

There's the (anticipated) problem that a-r-p-i creates a pipeline instance for cwl-runner, then cwl-runner goes and creates its own pipeline instance because it doesn't know about the 1st one. The pending solution is [#8876](#) and updating "components" of job object.

#18 - 05/11/2016 06:01 PM - Brett Smith

- Subject changed from [CWL SDK] Tool to create/update an Arvados pipeline template from a workflow to [CWL SDK] cwl-runner converts Keep references to CWL references

- Assigned To deleted (Tom Clegg)

- Target version changed from 2016-05-11 sprint to 2016-05-25 sprint

- Story points deleted (1.0)

#19 - 05/11/2016 07:17 PM - Peter Amstutz

- Story points set to 0.5

#20 - 05/11/2016 07:19 PM - Peter Amstutz

- Assigned To set to Peter Amstutz

#21 - 05/11/2016 07:35 PM - Tom Clegg

- Subject changed from [CWL SDK] cwl-runner converts Keep references to CWL references to [CWL SDK] cwl-runner can run a pipeline instance that was submitted by Workbench using a template from "cwl-runner --create-template"

#22 - 05/19/2016 02:31 AM - Peter Amstutz

Notes on 8653-cwl-runner-handle-files:

This branch became a bit bigger than originally intended. In addition to fixing cwl-runner to correctly handle files references provided by workbench, I chose to fix a couple of issues with command line argument processing, specifically:

- arvdos-cwl-runner inherited command line options from cwltool, even ones that didn't make sense and didn't work
- arvdos-cwl-runner --create-template required both a CWL file and an input object to be specified, even though the purpose was to create a workbench template the enabled the user to provide inputs.

I also bumped the cwltool version pin and did some related refactoring and code cleanup in cwltool, which affected APIs used by arvdos-cwl-runner.

#23 - 05/19/2016 02:31 PM - Radhika Chippada

- sdk/cwl tests are failing. 5 failures and 2 errors
- arvdos_cwl/__init__.py: arg_parser
 - Can you please add help message to basedir
 - is there a default value for eval-timeout? If so, would be desirable to add this detail in help message for this arg
 - project-uuid: in the help message, might be helpful to clarify that if not provided, work will be saved in the user's home project
- Just curious why the latter is better in the following:
 - document_loader, __, __ = cwltool.process.get_schema()
 - + document_loader = Loader({"cwl": "https://w3id.org/cwl/cwl#", "id": "@id"})

#24 - 05/19/2016 05:56 PM - Peter Amstutz

Radhika Chippada wrote:

- sdk/cwl tests are failing. 5 failures and 2 errors

Fixed, thanks for noticing.

- arvdos_cwl/__init__.py: arg_parser
 - Can you please add help message to basedir
 - is there a default value for eval-timeout? If so, would be desirable to add this detail in help message for this arg
 - project-uuid: in the help message, might be helpful to clarify that if not provided, work will be saved in the user's home project

Done.

- Just curious why the latter is better in the following:
 - document_loader, __, __ = cwltool.process.get_schema()
 - + document_loader = Loader({"cwl": "https://w3id.org/cwl/cwl#", "id": "@id"})

The reason I changed it is that get_schema() was changed to required a document version. However on further consideration I updated it to use cwltool.load_tool.fetch_document() which extracts the document version, so thanks for pointing that out.

#25 - 05/19/2016 06:14 PM - Radhika Chippada

- Thanks for updating. The help messages are much better now.
- The tests are still failing. 4 failures and 2 errors now. Making progress :)
- "on further consideration I updated it to use cwltool.load_tool.fetch_document() which extracts the document version, so thanks for pointing that out." In the future, I'll just say "what is this? i don't like it" :)

#26 - 05/19/2016 07:50 PM - Peter Amstutz

Tests should pass now @ [08cc91f](#)

#27 - 05/19/2016 07:59 PM - Radhika Chippada

LGTM

- Tests are passing now. I only ran sdk/cwl tests.

- I wondered about this but couldn't quite formulate a question around it. Thanks for updating this.

```
- if arvargs.create_template:  
+ if arvargs.create_template and not arvargs.job_order:
```

#28 - 05/19/2016 08:05 PM - Peter Amstutz

- *Status changed from In Progress to Resolved*
- *% Done changed from 60 to 100*

Applied in changeset arvados|commit:7a430017d745cab4458aa03f620e7925a50b7d06.