

## Arvados - Feature #8128

### [Crunch2] API support for crunch-dispatch

01/06/2016 08:37 PM - Tom Clegg

<b>Status:</b> Resolved	<b>Start date:</b> 04/28/2016
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assigned To:</b> Tom Clegg	<b>% Done:</b> 100%
<b>Category:</b>	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b> 2016-05-11 sprint	
<b>Description</b>	
See <a href="#">Container dispatch</a> ("Arvados API support")	
Some of the required APIs don't exist yet:	
<ul style="list-style-type: none"><li>"Locked" state</li><li>auth_uuid field: created automatically when container changes to state="Running"</li><li>arvados.v1.containers.auth(uuid=container.uuid): retrieve the api_client_authentication record corresponding to auth_uuid</li><li>api_client_authorizations.current: retrieve UUID of current token (similar to users.current). This avoids having to put the secret token anywhere other than the Authorization header.</li></ul>	
We do have:	
<ul style="list-style-type: none"><li>Filter containers list by API token: containers.list?filters=[[locked_by_uuid,=,my_token_uuid],[state,in,[Running,Locked]]]</li><li>Get list of queued containers: containers.list?filters=[[state,=,Queued]]</li><li>Get UUID of current token: api_client_authorizations.list?filters=[[api_token,=,my_token]]</li></ul>	
<b>Subtasks:</b>	
Task # 9106: Add "Locked" state	<b>Resolved</b>
Task # 9109: Add containers.container_auth API	<b>Resolved</b>
Task # 9098: Review 8128-crunch2-auth-api	<b>Resolved</b>
Task # 9107: Update container_auth_uuid as needed when state changes	<b>Resolved</b>
Task # 9108: Add api_client_authorizations.current API	<b>Resolved</b>
<b>Related issues:</b>	
Related to Arvados - Feature #6518: [Crunch] [Crunch2] Dispatch containers vi...	<b>Resolved</b> 07/08/2015
Related to Arvados - Feature #8028: [Crunch2] Dispatch containers locally	<b>Resolved</b> 01/19/2016
Related to Arvados - Bug #9900: [Crunch2] [API] Add ephemeral "run token" for...	<b>Resolved</b> 08/31/2016
Blocked by Arvados - Bug #8079: [API] Add uuid property to ApiClientAuthoriza...	<b>Resolved</b> 02/12/2016
Precedes (1 day) Arvados - Story #9187: [Crunch2] Update dispatcher to use ne...	<b>Resolved</b> 05/02/2016

#### Associated revisions

##### Revision 3ede205b - 05/12/2016 03:34 PM - Tom Clegg

Merge branch '8128-crunch2-auth-api'

closes #8128

#### History

##### #1 - 01/06/2016 08:44 PM - Tom Clegg

- Description updated

##### #2 - 01/22/2016 05:11 AM - Tom Clegg

- Description updated

##### #3 - 03/15/2016 06:12 PM - Tom Clegg

- Description updated

##### #4 - 04/27/2016 07:59 PM - Tom Clegg

- Target version set to 2016-05-11 sprint

#### #5 - 04/27/2016 08:21 PM - Tom Clegg

- Assigned To set to Tom Clegg

#### #6 - 04/28/2016 03:04 PM - Tom Clegg

- Description updated

#### #7 - 05/10/2016 01:54 PM - Tom Clegg

- Description updated

#### #8 - 05/10/2016 08:41 PM - Peter Amstutz

crunch-dispatch-local.go:

```
// Update container status to Running
err := arv.Update("containers", uuid,
    arvosclient.Dict{
        "container": arvosclient.Dict{"state": "Locked"}},
    nil)
if err != nil {
    log.Printf("Error updating container %v to 'Locked' state: %v", uuid, err)
}
```

- The comment says the 1st step is to "set container state to locked", but it locks the container record **after** starting the crunch-run -process. I believe the comment is correct and the code is wrong in this case.
- If it fails to lock the container, it logs an error but continues to run it anyway
- There's nothing here that fetches the container-specific authorization token after it is locked

crunch-dispatch-slurm.go:

- The comment says the 1st step is to "set container state to locked", but it locks the container record **after** submitting it to the slurm queue. I believe the comment is correct and the code is wrong in this case.
- Line 161 ends at column 237, which is a bit excessive even for my 200-character-wide Emacs window.
- There's nothing here that fetches the container-specific authorization token after it is locked

To be continued..

#### #9 - 05/11/2016 01:10 PM - Peter Amstutz

container.rb:

- Hard time following logic in validate\_lock, could use some comments.
- Need row-level locking (with\_lock) anytime container state field changes. Easy way to do that with Rails?

Current design seems to be that crunch-run is responsible for fetching auth token, but crunch-run isn't doing that yet. Crunch-run itself currently runs with same token as dispatcher. This seems like a bad idea, since the dispatcher token is a long-lived admin token.

Suggest that crunch-run runs with auth token (requires changes to dispatchers, no change to crunch-run). This would imply that both locked\_by and auth token are permitted to update container record.

Dispatchers should honor priority list: "order by priority DESC, modified\_at ASC"  
Also filter priority > 0

Slurm dispatcher should pass priority on sbatch command line, --priority=<value>

When claiming a Locked container previously owned by me, should check queue to make sure the job exists, clean up record if not.

#### #10 - 05/11/2016 01:19 PM - Peter Amstutz

Perhaps we can tweak the containers controller so that all updates take a row level lock?

#### #11 - 05/11/2016 03:29 PM - Tom Clegg

Added row-level lock in container controller.

Added comments in validate\_lock.



