Arvados - Bug #7444

[Crunch] Docker container not removed when job canceled, filling disk

10/02/2015 01:59 PM - Brett Smith

Status: Resolved
Priority: Normal
Assigned To: Tom Clegg
Category: Crunch
Target version: 2015-11-11 sprint

Start date: 10/02/2015
Due date: 
% Done: 100%
Estimated time: 0.00 hour

Description
We use docker run --rm to ensure that Docker containers are removed after tasks are finished, to prevent compute nodes from filling up with unused volumes. However, docker run --rm is handled by the Docker client. It simply makes the necessary API calls to remove the container after it exits.

Crunch's cancel code kills the Docker client. If a user cancels a job, the container will hang around, along with its volumes. We just had a situation where compute nodes on a cluster filled their /tmp partitions, because a user was canceling many jobs, leaving it full of finished Docker containers and their large tmp volumes.

Make sure that when Crunch cancels a job, the corresponding Docker container is removed.

Implementation

- Extend crunch-job to stop using --rm, and name containers after the task. Append "$try_number" to the name to avoid name collisions when tasks are retried.
- Extend the Docker cleaner service to listen for container stop events, and immediately destroy those containers. Sysadmins who want to debug Docker on compute nodes are expected to stop the Docker cleaner service to do that.

Subtasks:

<table>
<thead>
<tr>
<th>Task #</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>7560</td>
<td>Remove unused containers in dockercleaner</td>
<td>Resolved</td>
</tr>
<tr>
<td>7686</td>
<td>Remove --rm flag in crunch-job</td>
<td>Resolved</td>
</tr>
<tr>
<td>7693</td>
<td>Warn in install guide that dockercleaner will remove all stopped container...</td>
<td>Resolved</td>
</tr>
<tr>
<td>7561</td>
<td>Testing</td>
<td>Resolved</td>
</tr>
<tr>
<td>7547</td>
<td>Review 7444-dockercleaner-containers</td>
<td>Resolved</td>
</tr>
</tbody>
</table>

Associated revisions

Revision 1d1c6de3 - 11/04/2015 05:19 AM - Tom Clegg
Merge branch '7444-dockercleaner-containers' closes #7444

History

#1 - 10/02/2015 02:35 PM - Brett Smith
- Subject changed from [Crunch] Job containers not removed consistently, filling disk to [Crunch] Docker container not removed when job canceled, filling disk
- Description updated

#2 - 10/06/2015 06:25 PM - Brett Smith
- Description updated
- Story points set to 2.0

#3 - 10/13/2015 06:28 PM - Brett Smith
- Target version changed from Arvados Future Sprints to 2015-10-28 sprint

#4 - 10/14/2015 07:18 PM - Peter Amstutz
- Assigned To set to Peter Amstutz

#5 - 10/16/2015 06:19 PM - Brett Smith
- Target version changed from 2015-10-28 sprint to Arvados Future Sprints
Naming containers sounds like a good idea anyway, but seems tangential. Unless dockercleaner is supposed to pay attention to the names, perhaps in order to exempt non-Crunch containers from automatic removal...?

Should dockercleaner also delete all stopped containers that are already present when it starts up? This would help keep a long-running (e.g., bare metal) worker node clean.

If/when we do add this, I think it should have a separate command line flag, to support a workflow like

1. Turn off dockercleaner
2. Run a job
3. Turn on dockercleaner --leave-existing-containers
4. Inspect the container left behind by the above job, but let subsequent jobs get cleaned up

Until then, there's "docker ps --filter status=exited --format {{.ID}} | xargs docker rm".

You are right it is not necessary for the dockercleaner changes. I previously had an implementation idea based on naming containers predictably and having crunch-job remove them. This is basically a remnant of that—there was still interest in naming as a debugging aid.

Should dockercleaner also delete all stopped containers that are present when it starts up?

I'm interested in ops' opinion on this but my vote is yes.

Both changes (dockercleaner and crunch-job) are now in 7444-dockercleaner-containers at 9b48b17

- Status changed from New to In Progress

review 9b48b17edd8a5e396e0c59edf9354079350256c
LGTM

- Status changed from In Progress to Resolved
- % Done changed from 80 to 100

Applied in changeset arvados|commit:1d1c6de3c842a33a57b7d469fdaaaa1b873433dc.