

## Arvados - Bug #6702

### [Node Manager] Retries forever when a node creation request times out, even though the node was created

07/22/2015 03:53 PM - Brett Smith

<b>Status:</b>	Resolved	<b>Start date:</b>	07/22/2015
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Peter Amstutz	<b>% Done:</b>	67%
<b>Category:</b>	Node Manager	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2016-02-03 Sprint		
<b>Description</b>			
Node Manager decides to bring up a node. First this happens:			
<pre>2015-07-22_14:48:15.48284 2015-07-22 14:48:15 arvnodeman.nodeup[13142] WARNING: Client error: The read operation timed out - waiting 1 seconds</pre>			
Node Manager sees the failure and decides to retry the request. But on subsequent tries, this is always the response:			
<pre>2015-07-22_14:48:18.92984 2015-07-22 14:48:18 arvnodeman.nodeup[13142] WARNING: Client error: u"The resource 'projects/curoverse-production/zones/us-central1-a/instances/compute-yp0s2tcidw77kp-su921' already exists" - waiting 2 seconds</pre>			
The server handled the first request fine, we just didn't get the response back. We need to recognize when this happens and continue the node setup process, rather than retrying infinitely.			
There might be a few ways to do this:			
<ul style="list-style-type: none"><li>• If the exception makes the problem easily identifiable, just catch it and move it.</li><li>• At least some of the clouds let you send along a request ID with the request to ensure idempotency. Adding this to our requests might make the response nicer. I'm not sure—this would need testing.</li><li>• If all else fails, you could periodically check for the existence of the desired node, at least when it has a predictable name.</li></ul>			
Steps to fix:			
If a cloud error is raised by <code>create_node()</code> on GCE, test if the cloud node exists. If so, return the cloud node record and proceed. If not, raise the original error.			
<b>Subtasks:</b>			
Task # 8267: Add check if node exists			<b>Resolved</b>
Task # 8268: Add test?			<b>Resolved</b>
Task # 8263: Review 6702-gce-node-create-fix			<b>Resolved</b>

#### Associated revisions

##### Revision 6570eec0 - 02/02/2016 04:31 PM - Peter Amstutz

Merge branch '6702-gce-node-create-fix' closes #6702

##### Revision 73812c5a - 02/03/2016 08:54 PM - Peter Amstutz

Merge branch '6702-gce-node-create-fix' closes #6702

#### History

##### #1 - 07/28/2015 05:17 PM - Brett Smith

We just saw this again, except the original error was different: Google responded, "The zone 'projects/projname/zones/us-central1-a' does not have enough resources available to fulfill the request. Try a different zone, or try again later."

##### #2 - 01/19/2016 07:22 PM - Peter Amstutz

- Description updated

##### #3 - 01/19/2016 07:23 PM - Peter Amstutz

- Story points set to 1.0

**#4 - 01/20/2016 08:44 PM - Peter Amstutz**

- Target version changed from Arvados Future Sprints to 2016-02-03 Sprint

**#5 - 01/20/2016 08:47 PM - Peter Amstutz**

- Assigned To set to Peter Amstutz

**#6 - 02/02/2016 04:11 PM - Tom Clegg**

Is there some reason `gce.ComputeNodeDriver.create_node()` needs a copy of the code from `BaseComputeNodeDriver.create_node()`, instead of calling `super()`?

Would this same logic wouldn't work with other cloud drivers? It seems like the bug can happen just as easily with other clouds, so it should be in `BaseComputeNodeDriver`, unless there's some reason not to...

Otherwise LGTM.

**#7 - 02/02/2016 04:20 PM - Peter Amstutz**

Tom Clegg wrote:

Is there some reason `gce.ComputeNodeDriver.create_node()` needs a copy of the code from `BaseComputeNodeDriver.create_node()`,

instead of calling `super()`?

Because it needs `kwargs['name']`, which would incur either more refactoring or calling `self.arvados_create_kwargs` twice (once in the `GCE.create_node()` and again in `super.create_node()`) and possibly getting different results.

Would this same logic wouldn't work with other cloud drivers? It seems like the bug can happen just as easily with other clouds, so it should be in `BaseComputeNodeDriver`, unless there's some reason not to...

When we discussed the ticket earlier, the instructions were to only make the change for GCE. The same logic is definitely valid for Azure but I'm a bit fuzzy on whether it's also valid for AWS (I don't know if the name on AWS is a strongly unique identifier).

Otherwise LGTM.

**#8 - 02/02/2016 04:28 PM - Peter Amstutz**

To follow up, I just checked the `libcloud` driver for EC2. The "name" field on the Node object set by `libcloud` is either the "Name" tag or the instance id. However, the "Name" tag isn't set until after node creation succeeded and we don't know the instance id because we never got a response, so we can't use the same logic as Azure and GCE.

**#9 - 02/02/2016 04:35 PM - Peter Amstutz**

- Status changed from New to Resolved

Applied in changeset `arvados|commit:6570eec0115d7973cce4df10857631cfe6bd11c5`.

**#10 - 02/02/2016 04:38 PM - Tom Clegg**

Surely better to refactor than to have copy-and-pasted code. But it's a moot point if we move this to `BaseComputeNodeDriver`, which it sounds like we should.

Seems like we just need a method ("`node_lookup`") that returns `True` if the named node exists, and raise `NotImplemented` in the EC2 driver since names seem a bit less tight there. Aside from that, the existing code looks generic already.