Arvados - Bug #8953

[Node manager] can not shut down nodes anymore

04/12/2016 09:56 PM - Ward Vandewege

Status:	Resolved	**Start date:**	04/13/2016
Priority:	Normal	**Due date:**	
Assigned To:	Brett Smith	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2016-04-27 sprint		

Description

Node manager version 0.1.20160410021132-1 (with the drain fix from #8799) appears to have a shutdown loop...

```
2016-04-12_21:55:42.54948 2016-04-12 21:55:42 ComputeNodeShutdownActor.01df952eda4b.compute-q74ljtuggvufc9l-su92l[46608] INFO: Draining SLURM node compute19
2016-04-12_21:55:42.63657 2016-04-12 21:55:42 ComputeNodeShutdownActor.01df952eda4b.compute-q74ljtuggvufc9l-su92l[46608] INFO: Waiting for SLURM node compute19 to drain
2016-04-12_21:55:42.71020 2016-04-12 21:55:42 ComputeNodeShutdownActor.01df952eda4b.compute-q74ljtuggvufc9l-su92l[46608] INFO: Shutdown cancelled: shutdown window closed.
2016-04-12_21:55:42.71048 2016-04-12 21:55:42 ComputeNodeShutdownActor.01df952eda4b.compute-q74ljtuggvufc9l-su92l[46608] INFO: finished
2016-04-12_21:55:43.14420 2016-04-12 21:55:43 pykka[46608] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid:58d90fdd-7dee-4219-9ebb-429da8a893ef)
2016-04-12_21:55:43.14471 2016-04-12 21:55:43 pykka[46608] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:58d90fdd-7dee-4219-9ebb-429da8a893ef)
2016-04-12_21:55:43.15100 2016-04-12 21:55:43 ComputeNodeShutdownActor.429da8a893ef.compute-csm26hx92t8pf8o-su92l[46608] INFO: Draining SLURM node compute20
2016-04-12_21:55:43.27655 2016-04-12 21:55:43 ComputeNodeShutdownActor.429da8a893ef.compute-csm26hx92t8pf8o-su92l[46608] INFO: Waiting for SLURM node compute20 to drain
2016-04-12_21:55:43.36943 2016-04-12 21:55:43 ComputeNodeShutdownActor.429da8a893ef.compute-csm26hx92t8pf8o-su92l[46608] INFO: Shutdown cancelled: shutdown window closed.
2016-04-12_21:55:43.36970 2016-04-12 21:55:43 ComputeNodeShutdownActor.429da8a893ef.compute-csm26hx92t8pf8o-su92l[46608] INFO: finished
2016-04-12_21:55:43.87084 2016-04-12 21:55:43 pykka[46608] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid:f1192c0a-dbad-4bd6-9a70-949cb086f5bf)
2016-04-12_21:55:43.87123 2016-04-12 21:55:43 pykka[46608] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:f1192c0a-dbad-4bd6-9a70-949cb086f5bf)
2016-04-12_21:55:43.87793 2016-04-12 21:55:43 ComputeNodeShutdownActor.949cb086f5bf.compute-1q9g2jmg5askowg-su92l[46608] INFO: Draining SLURM node compute36
2016-04-12_21:55:44.08007 2016-04-12 21:55:44 ComputeNodeShutdownActor.949cb086f5bf.compute-1q9g2jmg5askowg-su92l[46608] INFO: Waiting for SLURM node compute36 to drain
2016-04-12_21:55:44.15134 2016-04-12 21:55:44 ComputeNodeShutdownActor.949cb086f5bf.compute-1q9g2jmg5askowg-su92l[46608] INFO: Shutdown cancelled: shutdown window closed.
2016-04-12_21:55:44.15164 2016-04-12 21:55:44 ComputeNodeShutdownActor.949cb086f5bf.compute-1q9g2jmg5askowg-su92l[46608] INFO: finished
2016-04-12_21:55:44.57382 2016-04-12 21:55:44 pykka[46608] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid:df6d159b-1837-4308-a409-e76748ab7dc6)
2016-04-12_21:55:44.57430 2016-04-12 21:55:44 pykka[46608] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:df6d159b-1837-4308-a409-e76748ab7dc6)
2016-04-12_21:55:44.58096 2016-04-12 21:55:44 ComputeNodeShutdownActor.e76748ab7dc6.compute-rs1ayc75z1atwki-su92l[46608] INFO: Draining SLURM node compute41
2016-04-12_21:55:44.69442 2016-04-12 21:55:44 ComputeNodeShutdownActor.e76748ab7dc6.compute-rs1ayc75z1atwki-su92l[46608] INFO: Waiting for SLURM node compute41 to drain
2016-04-12_21:55:44.75848 2016-04-12 21:55:44 ComputeNodeShutdownActor.e76748ab7dc6.compute-rs1ayc75z1atwki-su92l[46608] INFO: Shutdown cancelled: shutdown window closed.
2016-04-12_21:55:44.75878 2016-04-12 21:55:44 ComputeNodeShutdownActor.e76748ab7dc6.compute-rs1ayc75z1atwki-su92l[46608] INFO: finished
2016-04-12_21:55:45.13360 2016-04-12 21:55:45 ComputeNodeMonitorActor.1dab685fdd8a.compute-u3ps8tsf7ygy0bw-su92l[46608] DEBUG: Cannot shut down because node is not idle.
2016-04-12_21:55:45.32131 2016-04-12 21:55:45 pykka[46608] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid:955f8a87-54e4-4c11-8dd5-d6359dd39c83)
2016-04-12_21:55:45.32186 2016-04-12 21:55:45 pykka[46608] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:955f8a87-54e4-4c11-8dd5-d6359dd39c83)
```

```
2016-04-12_21:55:45.32826 2016-04-12 21:55:45 ComputeNodeShutdownActor.d6359dd39c83.compute-iaiguf1czbby51n-su92l[46608] INFO: Draining SLURM node compute30
2016-04-12_21:55:45.41042 2016-04-12 21:55:45 ComputeNodeShutdownActor.d6359dd39c83.compute-iaiguf1czbby51n-su92l[46608] INFO: Waiting for SLURM node compute30 to drain
2016-04-12_21:55:45.47984 2016-04-12 21:55:45 ComputeNodeShutdownActor.d6359dd39c83.compute-iaiguf1czbby51n-su92l[46608] INFO: Shutdown cancelled: shutdown window closed.
2016-04-12_21:55:45.48136 2016-04-12 21:55:45 ComputeNodeShutdownActor.d6359dd39c83.compute-iaiguf1czbby51n-su92l[46608] INFO: finished
2016-04-12_21:55:45.94573 2016-04-12 21:55:45 pykka[46608] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid:95f04d93-2173-4214-a811-216cc861cf67)
```

This gets repeated over and over and over and over again.

A snapshot of sinfo:

```
 sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute*     up   infinite     67 drain* compute[2,11,16,28,31,34,38-39,48,50,54-55,57-58,60,65,67-68,70,78,97-99,101,104-106,109,115,119,122,124,126,129-131,133-136,138,140-144,147,149-151,153-157,162,172-179,181-183]
compute*     up   infinite     74  down* compute[47,158,184-255]
compute*     up   infinite     12  drain compute[87,89-90,94-96,110,113,120,125,127,170]
compute*     up   infinite     57  alloc compute[0-1,3-5,7-10,12-14,17-18,21-27,29,33,43-44,46,61,63-64,66,72-74,76-77,81-82,84-86,88,91-93,100,102-103,107-108,111-112,114,116-118,121,123]
compute*     up   infinite     37   idle compute[6,15,19-20,30,32,35-37,40-42,45,49,51-53,56,59,62,69,71,75,79-80,83,132,137,139,146,148,152,161,163,167,171,180]
compute*     up   infinite      9   down compute[128,145,159-160,164-166,168-169]
crypto       up   infinite     67 drain* compute[2,11,16,28,31,34,38-39,48,50,54-55,57-58,60,65,67-68,70,78,97-99,101,104-106,109,115,119,122,124,126,129-131,133-136,138,140-144,147,149-151,153-157,162,172-179,181-183]
crypto       up   infinite     74  down* compute[47,158,184-255]
crypto       up   infinite     12  drain compute[87,89-90,94-96,110,113,120,125,127,170]
crypto       up   infinite     57  alloc compute[0-1,3-5,7-10,12-14,17-18,21-27,29,33,43-44,46,61,63-64,66,72-74,76-77,81-82,84-86,88,91-93,100,102-103,107-108,111-112,114,116-118,121,123]
crypto       up   infinite     37   idle compute[6,15,19-20,30,32,35-37,40-42,45,49,51-53,56,59,62,69,71,75,79-80,83,132,137,139,146,148,152,161,163,167,171,180]
crypto       up   infinite      9   down compute[128,145,159-160,164-166,168-169]
```

I'm ~~rolling su92l back to the previous release~~ upgrading node manager to version 0.1.20160412145128 and will report back on this ticket.

Subtasks:

Task # 8957: Review 8953-node-manager-prevent-shutdown-eligible-flapping-wip **Resolved**

Related issues:

Related to Arvados - Story #8000: [Node Manager] Shut down nodes in SLURM 'do... **Resolved**

Associated revisions

Revision 17c23d33 - 04/15/2016 08:18 PM - Peter Amstutz

Merge branch '8953-node-manager-FSM' closes #8953

Revision dfdb6060 - 04/16/2016 02:20 AM - Peter Amstutz

Don't try to drain node if no nodeename associated. refs #8953

Revision fe8143cc - 04/16/2016 02:31 AM - Peter Amstutz

Don't issue drain when shutdown has been cancelled. refs #8953

Revision 29379bea - 04/19/2016 08:05 PM - Peter Amstutz

Don't double-count nodes that are shutting down. refs #8953

Revision 9b90fe97 - 04/20/2016 01:10 PM - Peter Amstutz

Merge branch '8953-no-double-count' refs #8953

Revision b2cfb1a8 - 04/20/2016 06:26 PM - Peter Amstutz

Don't shut down if state is ('down', 'closed', 'boot wait', *) refs #8953

Revision 469102b3 - 04/21/2016 06:02 PM - Peter Amstutz

Fix race conditions in test_node_undrained_when_shutdown_cancelled
and test_boot_new_node_when_all_nodes_busy. refs #8953

History

#1 - 04/12/2016 10:00 PM - Ward Vandewege

- Description updated

#2 - 04/12/2016 10:02 PM - Ward Vandewege

- Description updated

#3 - 04/13/2016 12:20 AM - Ward Vandewege

version arvados-node-manager 0.1.20160412145128-1, same problem:

```
2016-04-13_00:15:44.35974 2016-04-13 00:15:44 ComputeNodeShutdownActor.689cbd7fea82.compute-7ig6hsrhv4ofmyc-su
92l[6938] INFO: Draining SLURM node compute61
2016-04-13_00:15:44.50012 2016-04-13 00:15:44 ComputeNodeShutdownActor.689cbd7fea82.compute-7ig6hsrhv4ofmyc-su
92l[6938] INFO: Waiting for SLURM node compute61 to drain
2016-04-13_00:15:44.71359 2016-04-13 00:15:44 ComputeNodeShutdownActor.689cbd7fea82.compute-7ig6hsrhv4ofmyc-su
92l[6938] INFO: Shutdown cancelled: shutdown window closed.
2016-04-13_00:15:44.71440 2016-04-13 00:15:44 ComputeNodeShutdownActor.689cbd7fea82.compute-7ig6hsrhv4ofmyc-su
92l[6938] INFO: finished
2016-04-13_00:15:45.30996 2016-04-13 00:15:45 pykka[6938] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid
:571dc061-b1e5-4109-8649-2c74a4de819d)
2016-04-13_00:15:45.31032 2016-04-13 00:15:45 pykka[6938] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:5
71dc061-b1e5-4109-8649-2c74a4de819d)
2016-04-13_00:15:45.31749 2016-04-13 00:15:45 ComputeNodeShutdownActor.2c74a4de819d.compute-lap4xwmilz8sp3o-su
92l[6938] INFO: Draining SLURM node compute9
2016-04-13_00:15:45.42928 2016-04-13 00:15:45 ComputeNodeShutdownActor.2c74a4de819d.compute-lap4xwmilz8sp3o-su
92l[6938] INFO: Waiting for SLURM node compute9 to drain
2016-04-13_00:15:45.53298 2016-04-13 00:15:45 ComputeNodeShutdownActor.2c74a4de819d.compute-lap4xwmilz8sp3o-su
92l[6938] INFO: Shutdown cancelled: shutdown window closed.
2016-04-13_00:15:45.53318 2016-04-13 00:15:45 ComputeNodeShutdownActor.2c74a4de819d.compute-lap4xwmilz8sp3o-su
92l[6938] INFO: finished
2016-04-13_00:15:46.45448 2016-04-13 00:15:46 pykka[6938] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid
:da445ca6-64dc-415b-b968-dcd70fd54145)
2016-04-13_00:15:46.45499 2016-04-13 00:15:46 pykka[6938] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:d
a445ca6-64dc-415b-b968-dcd70fd54145)
2016-04-13_00:15:46.46149 2016-04-13 00:15:46 ComputeNodeShutdownActor.dcd70fd54145.compute-qfjlcvisfspkeoh-su
92l[6938] INFO: Draining SLURM node compute64
2016-04-13_00:15:46.59047 2016-04-13 00:15:46 ComputeNodeShutdownActor.dcd70fd54145.compute-qfjlcvisfspkeoh-su
92l[6938] INFO: Waiting for SLURM node compute64 to drain
2016-04-13_00:15:46.71355 2016-04-13 00:15:46 ComputeNodeShutdownActor.dcd70fd54145.compute-qfjlcvisfspkeoh-su
92l[6938] INFO: Shutdown cancelled: shutdown window closed.
2016-04-13_00:15:46.71424 2016-04-13 00:15:46 ComputeNodeShutdownActor.dcd70fd54145.compute-qfjlcvisfspkeoh-su
92l[6938] INFO: finished
2016-04-13_00:15:46.80258 2016-04-13 00:15:46 JobQueueMonitorActor.140030151293568[6938] DEBUG: sending reques
t
2016-04-13_00:15:46.81523 2016-04-13 00:15:46 ArvadosNodeListMonitorActor.140030228209872[6938] DEBUG: sending
 request
2016-04-13_00:15:46.85562 2016-04-13 00:15:46 JobQueueMonitorActor.140030151293568[6938] DEBUG: Calculated wis
hlist: (empty)
2016-04-13_00:15:46.85607 2016-04-13 00:15:46 JobQueueMonitorActor.140030151293568[6938] INFO: got response wi
th 0 items in 0.0584580898285 seconds, next poll at 2016-04-13 00:15:56
2016-04-13_00:15:47.15560 2016-04-13 00:15:47 pykka[6938] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid
:0b1dc6f5-d357-45b8-b9c2-402620819306)
2016-04-13_00:15:47.15603 2016-04-13 00:15:47 pykka[6938] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:0
b1dc6f5-d357-45b8-b9c2-402620819306)
2016-04-13_00:15:47.16245 2016-04-13 00:15:47 ComputeNodeShutdownActor.402620819306.compute-miw9qgxcfg5ltcc-su
92l[6938] INFO: Draining SLURM node compute71
2016-04-13_00:15:47.30586 2016-04-13 00:15:47 ComputeNodeShutdownActor.402620819306.compute-miw9qgxcfg5ltcc-su
92l[6938] INFO: Waiting for SLURM node compute71 to drain
2016-04-13_00:15:47.44910 2016-04-13 00:15:47 ComputeNodeShutdownActor.402620819306.compute-miw9qgxcfg5ltcc-su
92l[6938] INFO: Shutdown cancelled: shutdown window closed.
2016-04-13_00:15:47.44942 2016-04-13 00:15:47 ComputeNodeShutdownActor.402620819306.compute-miw9qgxcfg5ltcc-su
92l[6938] INFO: finished
2016-04-13_00:15:50.11327 2016-04-13 00:15:50 pykka[6938] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid
:eecf8684-5c0d-481f-abee-0b6059da5133)
2016-04-13_00:15:50.11655 2016-04-13 00:15:50 pykka[6938] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:e
ecf8684-5c0d-481f-abee-0b6059da5133)
2016-04-13_00:15:50.43192 2016-04-13 00:15:50 ComputeNodeMonitorActor.7a73f10b2b4c.compute-lap4xwmilz8sp3o-su9
2l[6938] DEBUG: Cannot shut down because node is not idle.
2016-04-13_00:15:50.63696 2016-04-13 00:15:50 ArvadosNodeListMonitorActor.140030228209872[6938] INFO: got resp
onse with 357 items in 3.76417613029 seconds, next poll at 2016-04-13 00:15:56
2016-04-13_00:15:50.64313 2016-04-13 00:15:50 ComputeNodeShutdownActor.0b6059da5133.compute-xx1txhyfjmxtf0f-su
92l[6938] INFO: Draining SLURM node compute15
2016-04-13_00:15:50.82296 2016-04-13 00:15:50 pykka[6938] DEBUG: Unregistered ComputeNodeShutdownActor (urn:uu
id:b6a72c8b-6441-4767-a271-3cc932622b15)
2016-04-13_00:15:50.82327 2016-04-13 00:15:50 pykka[6938] DEBUG: Stopped ComputeNodeShutdownActor (urn:uuid:b6
a72c8b-6441-4767-a271-3cc932622b15)
2016-04-13_00:15:50.86064 2016-04-13 00:15:50 ComputeNodeShutdownActor.0b6059da5133.compute-xx1txhyfjmxtf0f-su
92l[6938] INFO: Waiting for SLURM node compute15 to drain
2016-04-13_00:15:50.97777 2016-04-13 00:15:50 ComputeNodeShutdownActor.0b6059da5133.compute-xx1txhyfjmxtf0f-su
92l[6938] INFO: Shutdown cancelled: shutdown window closed.
2016-04-13_00:15:50.97807 2016-04-13 00:15:50 ComputeNodeShutdownActor.0b6059da5133.compute-xx1txhyfjmxtf0f-su
92l[6938] INFO: finished
2016-04-13_00:15:51.76423 2016-04-13 00:15:51 pykka[6938] DEBUG: Registered ComputeNodeShutdownActor (urn:uuid
:0c01eaa1-4c64-4e5c-9bba-88938f54e8c4)
2016-04-13_00:15:51.76472 2016-04-13 00:15:51 pykka[6938] DEBUG: Starting ComputeNodeShutdownActor (urn:uuid:0
c01eaa1-4c64-4e5c-9bba-88938f54e8c4)
2016-04-13_00:15:51.77102 2016-04-13 00:15:51 ComputeNodeShutdownActor.88938f54e8c4.compute-xllo8zil1dbwgtx-su
92l[6938] INFO: Draining SLURM node compute75
2016-04-13_00:15:51.90690 2016-04-13 00:15:51 ComputeNodeShutdownActor.88938f54e8c4.compute-xllo8zil1dbwgtx-su
92l[6938] INFO: Waiting for SLURM node compute75 to drain
2016-04-13_00:15:52.15664 2016-04-13 00:15:52 ComputeNodeShutdownActor.88938f54e8c4.compute-xllo8zil1dbwgtx-su
92l[6938] INFO: Shutdown cancelled: shutdown window closed.
2016-04-13_00:15:52.15695 2016-04-13 00:15:52 ComputeNodeShutdownActor.88938f54e8c4.compute-xllo8zil1dbwgtx-su
92l[6938] INFO: finished
```

Status view:

```
#  sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute*     up   infinite     67 drain* compute[2,11,16,28,31,34,38-39,48,50,54-55,57-58,60,65,67-68,70,78,97
-99,101,104-106,109,115,119,122,124,126,129-131,133-136,138,140-144,147,149-151,153-157,162,172-179,181-183]
compute*     up   infinite     74  down* compute[47,158,184-255]
compute*     up   infinite     12  drain compute[87,89-90,94-96,110,113,120,125,127,170]
compute*     up   infinite     30  alloc compute[3-4,12-13,17-18,22-23,25,27,73-74,81-82,84-85,91-92,102-103,1
07-108,111-112,114,116-118,121,123]
compute*     up   infinite     64   idle compute[0-1,5-10,14-15,19-21,24,26,29-30,32-33,35-37,40-46,49,51-53,5
6,59,61-64,66,69,71-72,75-77,79-80,83,86,88,93,100,132,137,139,146,148,152,161,163,167,171,180]
compute*     up   infinite      9   down compute[128,145,159-160,164-166,168-169]
crypto       up   infinite     67 drain* compute[2,11,16,28,31,34,38-39,48,50,54-55,57-58,60,65,67-68,70,78,97
-99,101,104-106,109,115,119,122,124,126,129-131,133-136,138,140-144,147,149-151,153-157,162,172-179,181-183]
crypto       up   infinite     74  down* compute[47,158,184-255]
crypto       up   infinite     12  drain compute[87,89-90,94-96,110,113,120,125,127,170]
crypto       up   infinite     30  alloc compute[3-4,12-13,17-18,22-23,25,27,73-74,81-82,84-85,91-92,102-103,1
07-108,111-112,114,116-118,121,123]
crypto       up   infinite     64   idle compute[0-1,5-10,14-15,19-21,24,26,29-30,32-33,35-37,40-46,49,51-53,5
6,59,61-64,66,69,71-72,75-77,79-80,83,86,88,93,100,132,137,139,146,148,152,161,163,167,171,180]
crypto       up   infinite      9   down compute[128,145,159-160,164-166,168-169]
```

I'm reverting to version 0.1.20160407213044-1

#4 - 04/13/2016 01:46 AM - Brett Smith

- Status changed from New to In Progress

- Assigned To set to Brett Smith

- Target version set to 2016-04-13 sprint

#5 - 04/13/2016 01:52 PM - Peter Amstutz

Reviewing 8953-node-manager-prevent-shutdown-eligible-flapping-wip @ eb11bb1

slurm.ComputeNodeMonitorActor.shutdown_eligible() checks SLURM explicitly because dispatch.ComputeNodeMonitorActor.shutdown_eligible() has this code:

```
        if self.in_state('idle'):
            return True
        else:
            return "node is not idle." 
```

The problem is that a node in "drain" state is reported as "down" by Arvados so this will cancel the shutdown with the message "node is not idle."

The one solution may be to eliminate slurm.ComputeNodeMonitorActor.shutdown_eligible() and instead adjust the policy in dispatch.ComputeNodeMonitorActor.shutdown_eligible() to include 'down' nodes:

```
        if self.in_state('idle', 'down'):
            return True
        else:
            return "node is not idle."
```

#6 - 04/13/2016 01:57 PM - Brett Smith

- Target version changed from 2016-04-13 sprint to 2016-04-27 sprint

#7 - 04/13/2016 03:59 PM - Brett Smith

Peter Amstutz wrote:

> The one solution may be to eliminate slurm.ComputeNodeMonitorActor.shutdown_eligible() and instead adjust the policy in dispatch.ComputeNodeMonitorActor.shutdown_eligible() to include 'down' nodes:

Doing this naively causes test_no_shutdown_missing and test_no_shutdown_running_broken to fail. Should the tests be changed? I think the answer is *probably* yes, but that means we're at least partially going back on #7286. I refuse to push a branch that just changes the shutdown policy--*again*--without any actual design discussion or anything like that.

Node Manager was originally written with the philosophy of "If you are not affirmatively sure that the node is OK to shut down, don't shut it down." Since then, we've had a slew of tickets saying "I wanted Node Manager to shut this node down and it didn't," representing a design change. But after however many handfuls of these, nobody can actually articulate what the current design should be, so we get in this mess where ops gets frustrated because Node Manager doesn't do what they want it to do, and the developers get frustrated (or at least I sure do) because the functional requirements aren't clear, so of *course* it doesn't do what ops wants.

Since apparently nobody likes the original design philosophy anymore, ops should articulate a full set of new rules when nodes should or shouldn't be shut down. It should consider the following facts about a node, which is everything that shutdown_eligible and its surrounds currently considers:

- The last time Node Manager got a fresh list of compute nodes from the cloud
- The last time Node Manager got a fresh list of node records from the Arvados API server
- How long the cloud node has been up
    - Whether or not it has been up for at least boot_fail_after time, i.e., it should've been able to ping Arvados by now
    - Whether or not the cloud node is in a "shutdown window," i.e., near the end of its billing cycle
- Whether or not the cloud node is in a "broken" state by the cloud's own logic (e.g., the "ERROR" node state on Azure)
- The last time the cloud node successfully pinged Arvados (note that this might be "never")
- The node's SLURM state. Consider all of idle, alloc, down, drain, drng, fail, and their * variants.

#8 - 04/13/2016 06:59 PM - Brett Smith

- Target version changed from 2016-04-27 sprint to Arvados Future Sprints

#9 - 04/13/2016 08:31 PM - Brett Smith

Current behaviors being kept:

- The API poll freshness checks
- Shutting down when a node fails to ping after boot_fail_after

Basic idea: separate out the decision to put a node in DRAINING state, and the decision to shut down the cloud node. Node Manager always shuts down nodes in the DRAINED state. (Also FAIL and DOWN?)

Admin should be able to manually put nodes into a "maintenance" state, which indicates to crunch-dispatch not to schedule new jobs and for node manager not to shut down the node. Implementation TBD.

When not to put a node in DRAINING state:

- when Arvados says the node is busy (crunch_worker_state "busy")
- when Arvados says the node is idle (crunch_worker_state "idle") and the node is not inside a shutdown window
- when node is in the initial boot_fail_after grace period
- (???) when node has pinged recently and node is "broken" (error or unknown state on Azure)

When to put a node in DRAINING state:

Drain means node should finish its work and then be shut down.

- Check that node has been idle for some grace period (based on a timestamp recorded by crunch dispatch). When node is idle, switch it to DRAINING.

When to shut down a node on the cloud:

- Node is in any state other "idle", "alloc", or "drng" shut it down!
- Node is in any "star" state, shut it down

#10 - 04/14/2016 03:36 PM - Nico César

to start talking about all possible states are something like:

```
>>> slurm = ['drain','draining','alloc','down','error','fail','unknow']
>>> libcloud = ['running','rebooting','terminates','pending','unknown','stopped','suspended','error','paused']
>>> nodemanager = ['booting','shuttingdown','unpaired','paired']
>>> import pprint
>>> pprint.pprint([(a,b,c) for a in slurm for b in libcloud for c in nodemanager] )
```

which some of them make no sense and should no be common at all ... but this is a "map of the lanscape" of all the state that nodemanager will have to make decisions about

#11 - 04/14/2016 06:20 PM - Peter Amstutz

Detailed proposal at https://dev.arvados.org/projects/arvados/wiki/Node_manager_policy_matrix

Implementation steps:

1. Add "maintenance" flag to node record or properties, set and cleared by sysadmin
2. Add "last became idle" timestamp to node record, gets set by
3. API reports "drng" and "maintenance" as "busy"
4. NodeMonitorActor separate "eligible for drain" and "eligible for shutdown", consider_shutdown fires on any state change.
5. Separate "drain" and "shutdown" actors

Also:

1. libcloud option to not query node status on Azure

#12 - 04/14/2016 06:31 PM - Brett Smith

Peter Amstutz wrote:

> Detailed proposal at https://dev.arvados.org/projects/arvados/wiki/Node_manager_policy_matrix

If Ward likes it, it looks good to me.

At that point, we should do a branch that updates Node Manager to bring it as close as possible to this policy without changing any other components or distinguishing START_DRAIN and START_SHUTDOWN. (If the lack of distinction means we should avoid taking action in some cases, fine.) Node Manager being broken in master makes future deployments (possibly to get other Node Manager bugfixes) super awkward, so getting that fixed sooner rather than later is a priority.

#13 - 04/14/2016 06:44 PM - Peter Amstutz

Brett Smith wrote:

> Peter Amstutz wrote:
>
> > Detailed proposal at https://dev.arvados.org/projects/arvados/wiki/Node_manager_policy_matrix
>
> If Ward likes it, it looks good to me.
>
> At that point, we should do a branch that updates Node Manager to bring it as close as possible to this policy without changing any other components or distinguishing START_DRAIN and START_SHUTDOWN. (If the lack of distinction means we should avoid taking action in some cases, fine.) Node Manager being broken in master makes future deployments (possibly to get other Node Manager bugfixes) super awkward, so getting that fixed sooner rather than later is a priority.

I think doing a partial node-manager-only implementation of the policy is feasible. A "drng" node will be reported as "down" by the current API server, but if we keep the current shutdown behavior it will wait for the node to be in "drain" state before going on to actual cloud shutdown. This policy also eliminates the feature of cancelling shutdowns; Nico is concerned that this will lead to (even more) node churn without the idle grace period feature, so follow on work probably needs to be prioritized.

#14 - 04/14/2016 06:52 PM - Brett Smith

Peter Amstutz wrote:

> This policy also eliminates the feature of cancelling shutdowns; Nico is concerned that this will lead to (even more) node churn without the idle grace period feature, so follow on work probably needs to be prioritized.

I'm not wild about that commitment. Is there any work that can be done in Node Manager to advance toward the desired shutdown policy without removing the shutdown canceling code?

One thing I've been reminded of in the meantime: the shutdown canceling feature isn't just about actually canceling the shutdown from a policy perspective, but also about aborting shutdown attempts that Node Manager now believes will never succeed. See, e.g., 053de78cd, which cancels shutdowns in cases where the cloud API no longer lists the underlying node. In isolation, at least, that still seems like desirable behavior.

That's not incompatible with the long-term plan here; it's the distinction between having cancel_shutdown, and using the _stop_if_window_closed decorator. But I wanted to flag it because I don't think it's been brought up in the conversation yet.

#15 - 04/14/2016 07:12 PM - Peter Amstutz

Brett Smith wrote:

> Peter Amstutz wrote:
>
>> This policy also eliminates the feature of cancelling shutdowns; Nico is concerned that this will lead to (even more) node churn without the idle grace period feature, so follow on work probably needs to be prioritized.
>
> I'm not wild about that commitment. Is there any work that can be done in Node Manager to advance toward the desired shutdown policy without removing the shutdown canceling code?
>
> One thing I've been reminded of in the meantime: the shutdown canceling feature isn't just about actually canceling the shutdown from a policy perspective, but also about aborting shutdown attempts that Node Manager now believes will never succeed. See, e.g., 053de78cd, which cancels shutdowns in cases where the cloud API no longer lists the underlying node. In isolation, at least, that still seems like desirable behavior.
>
> That's not incompatible with the long-term plan here; it's the distinction between having cancel_shutdown, and using the _stop_if_window_closed decorator. But I wanted to flag it because I don't think it's been brought up in the conversation yet.

You're right, I'm conflating those two behaviors. We can remove the _stop_if_window_closed decorator without removing the cancel feature entirely.

#16 - 04/15/2016 02:29 PM - Peter Amstutz

- Target version changed from Arvados Future Sprints to 2016-04-27 sprint

8953-node-manager-FSM ready for review.

#17 - 04/15/2016 07:26 PM - Nico César

I kick the tests here https://ci.curoverse.com/job/developer-test-job/100/console

if they pass... LGTM!

#18 - 04/15/2016 08:04 PM - Ward Vandewege

Peter Amstutz wrote:

> Detailed proposal at https://dev.arvados.org/projects/arvados/wiki/Node_manager_policy_matrix

I finally looked, sorry for the delay. The only request I have is to treat alloc* and draining* not as 'to be shut down' immediately, because transient slurm issues can put nodes that are running jobs into those states.

#19 - 04/15/2016 08:25 PM - Peter Amstutz

- Status changed from In Progress to Resolved

- % Done changed from 0 to 100

Applied in changeset arvados|commit:17c23d338518f0498fb1396f24954f884a06b05b.