Arvados - Story #13697

Prevent the API server and database from continuing to serve requests to clients after timeout

06/28/2018 04:23 PM - Joshua Randall

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	0%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2021-09-29 sprint		

Description

A typical nginx configuration for the API server is to time out requests after 5m (300s). However, passenger/rails currently do not notice that the client has given up on the request and continue to process it. On some occasions (when things go wrong in other ways) we have had requests running for hours, holding locks that prevent other parts of the system from working, and in the worst cases, effecting a DoS on the entire API server by consuming all available passenger workers in the process.

There is no point in processing any part of a request for longer than the request timeout. An easy win to prevent the above scenario would be to set a postgres statement timeout to the same length as the nginx gateway timeout.

Our system has been running with a 300s statement_timeout for a few weeks without issue:
https://github.com/wtsi-hgi/arvados/commit/d9728e17148db53caf1f16fce032448c3d5c1432

Probably the value used for the timeout should come from config rather than being hard-coded, so that admins can configure it appropriately when non-standard nginx configuration is used.

Subtasks:

Task # 18156: Review	**New**

History

#1 - 02/20/2019 01:49 PM - Tom Morris

- Target version set to To Be Groomed

#2 - 03/06/2019 02:36 PM - Tom Morris

- Story points set to 0.5

#3 - 03/20/2019 02:19 PM - Tom Morris

- Target version changed from To Be Groomed to Arvados Future Sprints

#4 - 07/06/2021 09:12 PM - Peter Amstutz

- Target version deleted (Arvados Future Sprints)

#5 - 08/19/2021 04:54 PM - Joshua Randall

I am once again running into problems with API server resource starvation due to long-running requests causing normal clients with reasonable timeouts and retry behaviour essentially denial-of-service attacking the server (with, in this case, large collection creation requests).

IMHO it is not possible to run a stable Arvados API server under heavy load without applying a patch like the one that was implemented at Sanger to introduce statement timeouts in the database to mitigate this issue (see: https://github.com/wtsi-hgi/hgi-systems/blob/master/ansible/roles/arvados-master/files/hgi-integration-1.1.4.20180723133344-8.patch). Apologies that the OP for this issue did not include the patch!

Definitions of "heavy load" may vary depending on the capabilities of the API server and database hosts (and passenger settings such as the number of workers), but based on prior experience, the sorts of things that may take a long time (longer than the timeout) and lead to client-retry DoS attacks are:
- attempts to create very large collections (I think the thing that takes a long time is validating manifest block signatures)
- attempts to read very large collections with signed manifests (unsigned manifests are fast)
- attempts to work with container requests with large inline input documents
- certain queries against the log table, when it has grown very large

#6 - 08/20/2021 02:31 PM - Peter Amstutz

- Target version set to 2021-09-29 sprint

#7 - 09/15/2021 03:08 PM - Peter Amstutz

- Assigned To set to Tom Clegg