

openQA Project - coordination #96263

coordination # 80142 (Blocked): [saga][epic] Scale out: Redundant/load-balancing deployments of openQA, easy containers, containers on kubernetes

[epic] Exclude certain Minion tasks from "Too many Minion job failures alert" alert

2021-07-29 10:00 - mkittler

Status:	Blocked	Start date:	2020-09-01
Priority:	Normal	Due date:	2021-12-17
Assignee:	okurz	% Done:	25%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			
Description			
There are certain Minion tasks which fail regularly but there's not much we can do about it.			
<ul style="list-style-type: none">• #99837 All OBS rsync related tasks: Such failing tasks seem to have no impact; apparently it is sufficient if it works on the next attempt or users are able to fix problems themselves.• #70774 save_needle tasks: The needles dir on OSD might just be misconfigured by the user, e.g. we recently had lots of failing jobs for a needles directory for a new distribution. The users were often able to fix the problem themselves after they see the error (which is directly visible when saving a needle).• #100503 finalize_job_results tasks: I am not sure about this one. If the job just fails due to a user-provided hook script it should not be our business. On the other hand, we also configure the hook script for the investigation ourselves and want to be informed about problems. (So likely we want to consider these failures after all.)• #99831 jobs failing with the result 'Job terminated unexpectedly (exit code: 0, signal: 15)': This problem is independent of the task but is of course seen much more often on task we spawn many jobs of (e.g. finalize_job_result tasks) and tasks with possibly long running jobs (e.g. limit_assets). I suppose the error just means the Minion worker was restarted as signal 15 is SIGTERM. Since such tasks are either not very important or triggered periodically we can likely just ignore those failed jobs.			
Instead of implementing this on the monitoring-level we could also change openQA's behavior so these jobs are not considered failing. This would allow for a finer distinction (e.g. jobs would still fail if there's an unhandled exception due to a real regression). The disadvantage would be that all openQA instances would be affected. However, that's maybe not a bad thing and we can make it always configurable.			
Subtasks:			
action # 70774: save_needle Minion tasks fail frequently			New
coordination # 99831: [epic] Better handle minion tasks failing with "Job terminated un...			Blocked
action # 103416: Better handle minion tasks failing with "Job terminated unexpectedly" ...			In Progress
action # 99837: configurable exclusion rules for /influxdb/minion			New
action # 100503: Identify all "finalize_job_results" failures and handle them (report t...			Resolved
Related issues:			
Related to openQA Project - action #96197: [alert] web UI: Too many Minion jo...		Resolved	2021-07-28
Related to openQA Infrastructure - action #98499: [alert] web UI: Too many Mi...		Resolved	2021-09-13
Related to openQA Infrastructure - action #70768: obs_rsync_run and obs_rsync...		Resolved	2020-09-01

History

#1 - 2021-07-29 10:20 - mkittler

- Description updated

#2 - 2021-07-29 10:50 - mkittler

- Related to action #96197: [alert] web UI: Too many Minion job failures alert size:M added

#3 - 2021-07-29 15:28 - okurz

- Project changed from openQA Project to openQA Infrastructure

- Target version set to future

#4 - 2021-09-13 08:53 - mkittler

- Related to action #70774: save_needle Minion tasks fail frequently added

#5 - 2021-09-13 08:53 - okurz

- Related to action #98499: [alert] web UI: Too many Minion job failures alert size:S added

#6 - 2021-09-13 09:01 - okurz

Since commit 60ba6e1

Author: Amryslu xiaojing.liu@suse.com

Date: Wed Jan 13 14:40:19 2021 +0800

Ignore specified failed minion jobs

See: <https://progress.opensuse.org/issues/70768>

it is possible to exclude certain minion tasks. With this it should be possible to exclude more just based on configuration

#7 - 2021-09-13 09:01 - mkittler

- Related to action #70768: obs_rsync_run and obs_rsync_update_builds_text Minion tasks fail frequently added

#8 - 2021-09-13 10:02 - mkittler

- Description updated

#9 - 2021-09-23 19:51 - okurz

- Target version changed from future to Ready

After another OSD deployment another big chunk of minion jobs failing with "Job terminated unexpectedly (exit code: 0, signal: 15)" which we should handle in a better way. Adding ticket to backlog

#10 - 2021-09-29 09:20 - okurz

- Priority changed from Normal to High

#11 - 2021-09-29 09:22 - okurz

[#97979](#) was another place where we found that our minion job alert can be improved hence bumping prio to "High".

#12 - 2021-09-30 09:30 - okurz

- Status changed from New to In Progress

- Assignee set to okurz

we tried to estimate but maybe I can help to clarify first what we want to achieve

#13 - 2021-10-01 04:10 - openqa_review

- Due date set to 2021-10-15

Setting due date based on mean cycle time of SUSE QE Tools

#14 - 2021-10-01 13:53 - mkittler

There are multiple points to be addressed. Maybe this should have been an epic. However, I agree with okurz's previous comment about getting started ([#96263#note-9](#)).

By the way, we're generating the data for InfluxDB via the openQA web application. This is *not* using direct SQL queries. I've recently checked the concrete code and Minion API we use so far. There seemed no easy way to do a filtered query via the Mojo API so one would have to loop through all failed jobs to filter jobs with the result Job terminated unexpectedly out. *Maybe* it is simpler and more performant to query Minion's database tables directly via PostgreSQL like we already do for other monitoring data.

#15 - 2021-10-06 08:58 - okurz

- Tracker changed from action to coordination

- Subject changed from Exclude certain Minion tasks from "Too many Minion job failures alert" alert to [epic] Exclude certain Minion tasks from "Too many Minion job failures alert" alert

- Description updated

#16 - 2021-10-06 09:27 - okurz

- Description updated

#17 - 2021-10-06 09:32 - okurz

- Description updated

#18 - 2021-10-07 04:10 - openqa_review

Setting due date based on mean cycle time of SUSE QE Tools

#19 - 2021-10-07 07:27 - okurz

- Description updated

- Status changed from In Progress to Blocked

created four subtasks. One is in our backlog currently. Setting to "Blocked" on subtasks.

#20 - 2021-10-12 10:46 - okurz

- Status changed from Blocked to New

- Assignee deleted (okurz)

- Target version changed from Ready to future

currently not to be considered for backlog due to backlog size-constraints

#21 - 2021-12-01 07:48 - okurz

- Project changed from openQA Infrastructure to openQA Project

- Category set to Feature requests

- Target version changed from future to Ready

- Parent task set to #80142

#22 - 2021-12-02 11:10 - okurz

- Status changed from New to Blocked

- Assignee set to okurz