

## openQA Project - coordination #99831

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

coordination # 96263 (New): [epic] Exclude certain Minion tasks from "Too many Minion job failures alert" alert

### [epic] Better handle minion tasks failing with "Job terminated unexpectedly"

2021-10-06 08:41 - okurz

<b>Status:</b> New	<b>Start date:</b> 2021-12-02
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 50%
<b>Category:</b> Feature requests	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b> future	
<b>Difficulty:</b>	
<b>Description</b>	
<b>Motivation</b>	
<p>Often we have alerts about "Too many Minion job failures". Some of them are 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 should review the different kind of jobs we have and decide if we should ignore those failures, turn them into "passed" (maybe need upstream feature) or handle somehow differently, e.g. retrigger automatically.</p> <p>Alerts for Too many Minion job failures with the result Job terminated unexpectedly (exit code: 0, signal: 15):</p> <ul style="list-style-type: none"><li>- This problem is seen when we spawn many jobs of (e.g. finalize_job_result tasks) and tasks with possibly long running jobs (e.g. limit_assets).</li><li>- The Minion worker was restarted as signal 15 is SIGTERM.</li></ul>	
<b>Acceptance criteria</b>	
<ul style="list-style-type: none"><li>• <b>AC1:</b> All our minion jobs have a sigterm handler to decide how to shut down in a clean way in a reasonable time</li><li>• <b>AC2:</b> Our minion job list on OSD and O3 do not show any "Job terminated unexpectedly" over multiple deployments</li></ul>	
<b>Suggestions</b>	
<ul style="list-style-type: none"><li>• Since such tasks are either not very important or triggered periodically we should review the different kind of jobs we have and decide if we should ignore those failures, turn them into "passed" (maybe need upstream feature) or handle somehow differently, e.g. retrigger automatically.</li><li>• Implement sigterm handler for each minion job</li><li>• Test on o3 and osd either by manually restarting openqa-gru multiple times or awaiting the result from multiple deployments and checking the minion dashboard, e.g. <a href="https://openqa.suse.de/minion/jobs?state=failed">openqa.suse.de/minion/jobs?state=failed</a></li></ul>	
<b>Subtasks:</b>	
action # 103416: Better handle minion tasks failing with "Job terminated unexpectedly" ...	<b>Resolved</b>
action # 104116: Better handle minion tasks failing with "Job terminated unexpectedly" ...	<b>Resolved</b>
action # 107533: Better handle minion tasks failing with "Job terminated unexpectedly" ...	<b>Resolved</b>
action # 108980: Better handle minion tasks failing with "Job terminated unexpectedly" ...	<b>New</b>
action # 108983: Better handle minion tasks failing with "Job terminated unexpectedly" ...	<b>New</b>
action # 108989: Better handle minion tasks failing with "Job terminated unexpectedly" ...	<b>New</b>

### History

#### #1 - 2021-11-10 10:56 - okurz

Discussed in 2021-11-10 as we currently again found some minion jobs failing with the above symptoms. Whenever processes are stopped or restarted it's likely that we hit an job\_finalize\_result job because we have many openQA jobs. These minion jobs likely only take some seconds so we should ensure that we have explicit TERM signal handling to either just finish the task or stop gracefully within a reasonable time, i.e. some seconds. Because the main minion job handler tries to bring down minion jobs gracefully we should be ok to just ignore the TERM signal.

#### #2 - 2021-11-10 11:45 - mkittler

Note that the default "KillMode" of systemd is control-group by default so SIGTERM is sent to all processes (see <https://www.freedesktop.org/software/systemd/man/systemd.kill.html#KillMode>). I suppose for a graceful termination of Minion jobs we needed to set it to mixed and possibly increase the time until SIGKILL is sent. Then the remaining timeouted jobs should be distinguishable as they've received SIGKILL (and not SIGTERM). Possibly the Minion framework could still be changed to make such cancelled jobs better distinguishable (and restart such jobs automatically depending on some setting of the job).

### #3 - 2021-12-01 07:48 - okurz

- Project changed from openQA Infrastructure to openQA Project

- Target version changed from future to Ready

As we again have multiple minion job alerts just after deploying OSD I am adding this to the backlog

### #4 - 2021-12-01 10:27 - cdywan

Indeed this is from this morning:

```
---
args: []
attempts: 1
children: []
created: 2021-11-30T23:00:05.58353Z
delayed: 2021-11-30T23:00:05.58353Z
expires: 2021-12-02T23:00:05.58353Z
finished: 2021-12-01T05:23:05.67069Z
id: 3538496
lax: 0
notes:
  gru_id: 30652445
parents: []
priority: 5
queue: default
result: 'Job terminated unexpectedly (exit code: 0, signal: 15)'
retried: ~
retries: 0
started: 2021-11-30T23:00:05.58784Z
state: failed
task: limit_results_and_logs
time: 2021-12-01T10:16:52.2364Z
worker: 575

> sudo grep --color=always limit_results_and_logs /var/log/apache2/error_log /var/log/apache2/error_log /var/log/openqa{,_gru}
/var/log/openqa_gru:[2021-12-01T00:00:05.597881+01:00] [debug] Process 31351 is performing job "3538496" with task "limit_results_and_logs"
```

### #5 - 2021-12-02 10:25 - okurz

- Tracker changed from action to coordination

- Subject changed from Better handle minion tasks failing with "Job terminated unexpectedly" to [epic] Better handle minion tasks failing with "Job terminated unexpectedly"

- Description updated

- Category set to Feature requests

### #6 - 2021-12-02 11:09 - okurz

- Status changed from New to Blocked

- Assignee set to okurz

### #7 - 2021-12-14 21:11 - okurz

- Status changed from Blocked to Feedback

With the one subtask resolved where mkittler also covered other jobs we can now check the situation after the next OSD deployments if there are minion jobs left with the symptom of "Job terminated unexpectedly". If all solved then we can go to a daily deployment or even higher frequency :)

### #8 - 2021-12-17 12:41 - mkittler

I was looking at the archiving code anyways (for [#104136](#)), so here a PR for the archiving task: <https://github.com/os-autoinst/openQA/pull/4415>

### #9 - 2021-12-20 07:07 - okurz

- Status changed from Feedback to Resolved

After another deployment <https://openqa.suse.de/minion/jobs?state=failed> actually shows no entries. This is great!

**#10 - 2022-02-24 13:46 - mkittler**

- Status changed from Resolved to Feedback

Today I looked into Minion jobs and found further instances, e.g. one archiving job and many finalize jobs. I don't think this is solved until all jobs use the signal guard (also see AC1).

**#11 - 2022-02-24 13:51 - okurz**

- Status changed from Feedback to Blocked

ok, fine. Created a specific ticket [#107533](#) for "job\_finalize\_result"

**#12 - 2022-03-25 10:15 - okurz**

- Status changed from Blocked to Workable

- Assignee deleted (okurz)

**#13 - 2022-03-25 13:17 - mkittler**

- Assignee set to mkittler

**#14 - 2022-03-25 13:28 - mkittler**

Remaining tasks which are still missing the signal handler:

1. OpenQA::Task::Asset::Download
2. OpenQA::Task::Iso::Schedule
3. OpenQA::Task::Needle::Delete
4. OpenQA::Task::Needle::Save

Maybe 3. and 4. aren't that important because concluding the Git operation shouldn't take long (and therefore not exceed the timeout anyways). At least continuing those shouldn't be worse than trying to cleanup what has already changed so far. We could of course skip the cleanup and hard-reset the Git repository as first step in these tasks (so it wouldn't matter if there are half-done changes). That would also help with <https://progress.opensuse.org/issues/70774>.

**#15 - 2022-03-25 13:51 - mkittler**

- Status changed from Workable to Blocked

Blocked by newly created subtasks.

**#16 - 2022-04-05 14:50 - mkittler**

There's still a small amount of finalize\_job\_results jobs (and even one archive\_job\_results job) failing with "Job terminated unexpectedly". So I suppose the small window before setting up the signal handler is in fact problematic. However, it makes likely more sense to work on the other sub tasks of the parent epic as these problems produce way more failed jobs now.

**#17 - 2022-07-06 10:55 - okurz**

- Status changed from Blocked to New

- Assignee deleted (mkittler)

- Target version changed from Ready to future

still blocked by subtasks but all are currently not in the backlog hence removing this ticket from the backlog as well