

openQA Project - action #80908

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

[epic] Continuous deployment (package upgrade or config update) without interrupting currently running openQA jobs

2020-12-09 10:09 - okurz

Status: New	Start date: 2020-12-09
Priority: Normal	Due date:
Assignee:	% Done: 100%
Category: Feature requests	Estimated time: 0.00 hour
Target version: future	
Difficulty:	

Description Motivation We want to upgrade more often but not disrupt openQA jobs on package upgrades as well as re-read configuration whenever a job finishes Acceptance criteria <ul style="list-style-type: none">• AC1: openQA worker packages can be upgraded continuously without interrupting currently running openQA jobs• AC2: openQA workers read updated configuration, e.g. WORKER_CLASS, whenever they are ready to pick up new jobs• AC3: Both o3 and osd deploy automatically after every change if all relevant checks have passed Ideas <ul style="list-style-type: none">• Use different git branches, e.g. "dev" or "main" and then "stable" or "tested" or "release" and create automatic merges by bots based on checks• Switch o3 workers to either deploy from worker containers which we update continuously or change the worker to allow non-transactional updates Further details <p>One could try what apache does with apache2ctl graceful or systemctl reload apache2, e.g. see https://elearning.wslsp.com/pcmagazine/apache-graceful-restart-centos-7/</p> <p>The restart of openQA workers could be simply prevented or delayed, e.g. with SendSIGKILL= in the openQA worker systemd service definitions which every openQA user is free to do, but then we could potentially wait hours until the service restarts if ever. Maybe we can still add a "graceful-stop" mode, wait a useful time for all jobs to finish and then restart (or even reboot the host).</p>	
Subtasks:	
action # 80910: openQA workers read updated configuration, e.g. WORKER_CLASS, whenever ...	Resolved
action # 80986: terminate worker process after executing all currently assigned jobs ba...	Resolved
action # 81884: openqa-webui should automatically restart on config updates	Resolved
action # 89200: Switch OSD deployment to two-daily deployment	Resolved
action # 90152: module results missing on quick job (on auto-restarting worker)	Resolved

History

#1 - 2020-12-09 10:25 - okurz

- Description updated

- Status changed from New to Blocked

- Assignee set to okurz

- Target version changed from future to Ready

I will track this epic as we already have one specific subtask. That should be good enough for now.

#2 - 2020-12-09 10:39 - okurz

- Description updated

#3 - 2020-12-09 16:19 - okurz

In <https://chat.suse.de/group/qa-tools?msg=cRrsekSpzTHxMPRoz> we discussed that maybe it can be as simple as "terminate after executing all currently assigned jobs" and let the worker be automatically restarted by systemd (or kubernetes). This way reading the config as well as reading any new files (after a package upgrade) would work.

#4 - 2020-12-10 10:50 - Xiaojing_liu

Discussed this with the Migration and Security teams, before this ticket is resolved, if the deployment time is the milestone release candidate day, could we postpone the OSD deployment? Because on that day, they need to give the test report ASSP, if the deployment stops and re-triggers their jobs, it will cost them more time to wait for the result. This is just a workaround before we improve the deployment progress.

#5 - 2020-12-11 10:54 - okurz

Xiaojing_liu wrote:

Discussed this with the Migration and Security teams, before this ticket is resolved, if the deployment time is the milestone release candidate day, could we postpone the OSD deployment? Because on that day, they need to give the test report ASSP, if the deployment stops and re-triggers their jobs, it will cost them more time to wait for the result. This is just a workaround before we improve the deployment progress.

I understand that. But we were at this point already some years ago and there was always something very important coming up blocking deployment. You brought up the idea in chat about shifting the deployment time. You could create a ticket for that and we can decide there.

#6 - 2020-12-14 15:23 - mkittler

Here's an idea to improve this further: <https://github.com/os-autoinst/openQA/pull/3641> (besides [#80986](#))

#7 - 2021-01-11 10:43 - okurz

- Subject changed from [epic] Continuous deployment without interrupting currently running openQA jobs to [epic] Continuous deployment (package upgrade or config update) without interrupting currently running openQA jobs

#8 - 2021-01-13 10:51 - mkittler

The restart of openQA workers could be simply prevented or delayed, e.g. with `SendSIGKILL=` in the openQA worker systemd service definitions which every openQA user is free to do, but then we could potentially wait hours until the service restarts if ever.

Because of the "but" I would refrain from following this approach. It would be a huge regression if restarting a worker is no longer possible in the way it worked before.

One could try what apache does with `apache2ctl graceful` or `systemctl reload apache2`

This is also what I had in mind with my "SIGHUB" approach and likely it is also the way to go as we've seen that simply reloading the config within the worker is [more complicated than expected](#) and only solves half of the epic.

With <https://github.com/os-autoinst/openQA/pull/3641> I implemented *almost* the Apache/NGINX behavior besides the fact that my change only allows to terminate the worker gracefully but won't start it again on its own. I suppose for that we'd needed a master process and at least one worker process so the master can start the worker again as needed. At least all applications I know which can restart itself have at least 2 processes. Since we always use systemd the idea was to simply use `Restart=always` via `openqa-worker-auto-restart@.service`, see <https://progress.opensuse.org/issues/80910#note-16>.

My PR suggests to use e.g. `systemctl kill --signal SIGHUP openqa-worker-auto-restart@*.service`. However, we could of course add `ExecReload=/bin/kill -HUP $MAINPID` to `openqa-worker-auto-restart@.service` to allow e.g. `systemctl reload openqa-worker-auto-restart@*.service`.

This still leaves it open how we'd like to trigger the "reload". I see three ways we can do that and these are not meant as different alternatives but as different ways which would complement each other:

1. Let systemd trigger the "reload" when the config file changes (draft: <https://github.com/os-autoinst/openQA/pull/3666>).
2. Add an RPM hook to invoke `systemctl kill --signal SIGHUP openqa-worker-auto-restart@*.service` on updates. I don't know how to do that but considering there are already existing hooks for restarting it should be possible. (The implementation for restarting can be found in `/usr/lib/rpm/macros.d/macros.systemd` provided by the `systemd-rpm-macros` package. The relevant macro which is also used in our spec file is `%service_del_postun` which in turn uses `%_restart_on_update()`.
 - If that's not possible we could also try to use salt but it will be the less generic approach. (draft SR: https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/423/diffs)
 - We also learned that the existing hook to restart will not go in our way as long as `openqa-worker.target` is not active. (see <https://progress.opensuse.org/issues/80910#note-16>)

- Draft: <https://github.com/os-autoinst/openQA/pull/3699>
3. Document how to trigger a graceful restart manually. To be more use-friendly, we could really go for setting ExecReload to allow using systemctl reload
- Done: https://open.qa/docs/#_stoppingrestarting_workers_without_interrupting_currently_running_jobs

I'd also like to note that if we're able to trigger it reliably, we could actually revert the use of Environment=OPENQA_WORKER_TERMINATE_AFTER_JOBS_DONE=1 so workers would not restart unnecessarily. Unnecessary restarts of the worker are obviously a negative "side-effect" of using openqa-worker-auto-restart@.service so far (considering the goal is to apply only updates and configuration changes). Note that support for the environment variable itself should stay because it might also be useful in other use cases, e.g. when running the worker within a container; I'm only suggesting to remove it from the systemd service. Maybe it would also be better to add yet another service instead of modifying the existing one in an incompatible way.

#9 - 2021-01-13 11:09 - mkittler

- Status changed from Blocked to Feedback

I'm setting this to feedback because I'd actually like to hear some feedback from the team before proceeding. (If the approach is accepted this would solve the 2 sub tasks in one go so I added this comment on the epic-level.)

#10 - 2021-01-19 09:26 - okurz

Your overall approach looks sound and safe.

mkittler wrote:

1. Document how to trigger a graceful restart manually. To be more use-friendly, we could really go for setting ExecReload to allow using systemctl reload

I guess this could be done without interfering with other functionality. I know we are struggling a bit with this epic to know where to go. How about you try to create a draft pull request that starts with the actual *documentation changes* starting from the users point of view based on the motivation and ACs for this epic and doing the implementation afterwards.

#11 - 2021-01-19 16:31 - mkittler

Ok, but it would really be a draft because it would not actually work until we switch to using openqa-worker-auto-restart@.service to have Restart=always. Enabling that service would be the next step then. I've already tested it on imagetester so the next step would be testing the salt change I've prepared on staging.

#12 - 2021-01-20 16:19 - mkittler

- Assignee changed from okurz to mkittler

- PR for documentation: <https://github.com/os-autoinst/openQA/pull/3682>
- The SR for using openqa-worker-auto-restart@.service seems to work on a staging worker (see comments in GitLab): https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/426

Since this first step is important for both sub tickets I'm writing this comment in the epic ticket and also assign it to myself.

#13 - 2021-01-27 12:28 - mkittler

- Documentation PR has been merged.
- The SR has been merged so openqa-worker-auto-restart@.service is now used on OSD workers and openqa-worker.target (which would cause restarts on package updates) is disabled.

#14 - 2021-01-28 14:13 - mkittler

Outstanding PRs:

- reload on RPM update: <https://github.com/os-autoinst/openQA/pull/3699>
- reload on config change: <https://github.com/os-autoinst/openQA/pull/3666>
- OSD config: https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/438

#15 - 2021-02-09 09:21 - cdywan

mkittler wrote:

Outstanding PRs:

- reload on RPM update: <https://github.com/os-autoinst/openQA/pull/3699>
- reload on config change: <https://github.com/os-autoinst/openQA/pull/3666>

- OSD config: https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/438

All M/PRs are merged now

#16 - 2021-02-10 10:26 - cdywan

As mentioned during the daily, a follow-up was also needed here: https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/446 (discussed and merged)

#17 - 2021-02-10 12:48 - mkittler

Doesn't work in production, see [#80910#note-25](#). I'm working on a fix. I stopped and masked openqa-reload-worker-auto-restart@*.path services on OSD workers to avoid further jobs failing with isotovideo received signal HUP.

#18 - 2021-02-24 11:34 - mkittler

After the deployment today everything seems to work:

- We've seen that the package update workers receive SIGHUP.
- I've been unmasking the paths services: salt -C 'G@roles:worker' cmd.run 'systemctl unmask openqa-reload-worker-auto-restart@{1..50}.path' (globbing not possible here)
- salt -C 'G@roles:worker' state.apply no longer complains about masked services which are no up and running.
- I've tested editing the worker config and manually reloading the worker service and both leads to the worker receiving SIGHUP.
- The worker behaves correctly when receiving SIGHUP while idling, while setting up a job and while running a job (also when receiving it twice). The jobs being executed while receiving SIGHUP pass normally. There are no further jobs like [#89056](#).

#19 - 2021-02-24 11:35 - mkittler

- Status changed from Feedback to Resolved

#20 - 2021-02-24 13:27 - okurz

Awesome. Would you be interested in demo that feature in the next SUSE QE Tools workshop?

#21 - 2021-02-26 09:07 - okurz

- Status changed from Resolved to Feedback

[mkittler](#) because I have not seen a reply from you to the above and because I think we can do more here, at least find follow-up tickets I am reopening. Maybe we can discuss in the weekly what to do next about it

#22 - 2021-02-26 11:10 - okurz

- Description updated

- Status changed from Feedback to Blocked

- Assignee changed from mkittler to okurz

#23 - 2021-06-28 11:50 - okurz

- Status changed from Blocked to New

- Assignee deleted (okurz)

- Target version changed from Ready to future