openQA Project - action #69355

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

[spike] redundant/load-balancing webui deployments of openQA

2020-07-25 15:16 - okurz

Status:	Resolved	**Start date:**	2020-07-25
Priority:	Low	**Due date:**	
Assignee:	ilausuch	**% Done:**	0%
Category:	Feature requests	**Estimated time:**	0.00 hour
Target version:	Ready		
Difficulty:			

Description

Motivation

Single instances of the webui can cause longer downtimes and make upgrades of OS more risky, e.g. when we do not have management access to VMs that might fail to reboot. Also, load-balancing can help as well as having switch-over deployments possible for easier testing, staging, etc.

Acceptance criteria

- **AC1:** a proof of concept exists, eg. a test instance setup that can be reproduced
- **AC2:** documentation exists how to setup redundant load-balancing infrastructures

Suggestions

As our worker setup already has good inherent redundancy and load-balancing start with the webui server

state-of-the-art is k8s so we should aim for that. Maybe a "docker compose" file is a good intermediate step, then k8s with a helm chart, potentially also some setup based on gitlab, see https://docs.gitlab.com/ee/ci/environments/incremental_rollouts.html#blue-green-deployment

Related issues:

Related to openQA Infrastructure - action #65154: root partition on osd excee...	**Resolved**	**2020-04-01**	**2020-08-04**
Related to openQA Project - action #55262: Install Pgpool-II or PgBouncer bef...	**New**	**2019-08-08**	
Related to openQA Project - action #43715: Update upstream dockerfiles to pro...	**Resolved**	**2018-11-13**	
Related to openQA Project - action #43712: Update upstream dockerfiles to pro...	**Resolved**	**2018-11-13**	
Related to openQA Project - action #41600: fallback mechanism for apache, e.g...	**New**	**2018-09-26**	
Copied to openQA Project - action #76990: Improve documentation for redundant...	**Resolved**		

History

#1 - 2020-07-25 15:17 - okurz

- Related to action #65154: root partition on osd exceeds alert threshold, 90%, after osd deployment -> apply automatic reboots to OSD machines added

#2 - 2020-07-30 09:46 - okurz

- Related to action #55262: Install Pgpool-II or PgBouncer before PostgreSQL for openQA instances, e.g. to be used on OSD added

#3 - 2020-08-27 10:51 - mkittler

> start with the webui server

The web UI server runs a lot of services and provides the filesystem /var/lib/openqa so this raises a lot of questions:

1. I assume there would still be just one database.
2. What about files within /var/lib/openqa? I suppose this file system still needs to be shared between the web UIs, e.g. using NFS like we already

share /var/lib/openqa/share with the worker hosts.
3. What about the scheduler? I suppose there should still be just one to avoid race conditions.
4. What about the liveviewhandler and websocket server? Both are so far separate services because only a single process is supposed to serve the routes they provide because they rely on resources which needed to be shared between multiple instances if there were multiple instances. In case of the liveviewhandler that resource is websocket connection to another server so sharing it would be quite hard. So there should be only one of these services (in their current form).
5. What about the Minion service? Since we rely on locking to avoid running jobs in parallel which shouldn't run in parallel I'd say there should only be one Minion service as well or multiple Minion services would share the same database. These jobs often involve a lot filesystem access under /var/lib/openqa so likely having multiple Minion services (sharing the same database) is not very beneficial if these filesystem access would then be performed from a remote host.

Looks like we would really only run multiple instances of the openqa-webui.service and keep the other services on the web UI host as they are. Currently we're actually running multiple web UI processes via preforking. Maybe we could reduce the number of prefork processes then. Otherwise we would likely need to take care not to exceed the limit for PostgreSQL connections, e.g. by implementing #55262.

> can cause longer downtimes and make upgrades of OS more risky, e.g. when we do not have management access to VMs that might fail to reboot.

But *some* host needs to provide the "entry point" and database, right? So if that host is down we would still have a downtime.

> Also, load-balancing can help as well as having switch-over deployments possible for easier testing, staging, etc.

As long as we share the same database I would be very careful to run multiple instances of different openQA versions in parallel.

#4 - 2020-09-09 11:22 - okurz

mkittler wrote:

> > start with the webui server
>
> The web UI server runs a lot of services and provides the filesystem /var/lib/openqa so this raises a lot of questions:
>
> 1. I assume there would still be just one database.

either that or database replication

> 1. What about files within /var/lib/openqa? I suppose this file system still needs to be shared between the web UIs, e.g. using NFS like we already share /var/lib/openqa/share with the worker hosts.

Yes, I do not see an immediate problem with that as we already have either scalability features in place, like minion jobs with locks, or the same problems we want to fix anyway, like when saving needles interferes with updating the git repo with fetchneedles.

> 1. What about the scheduler? I suppose there should still be just one to avoid race conditions.

yes

> 1. What about the liveviewhandler and websocket server? Both are so far separate services because only a single process is supposed to serve the routes they provide because they rely on resources which needed to be shared between multiple instances if there were multiple instances. In case of the liveviewhandler that resource is websocket connection to another server so sharing it would be quite hard. So there should be only one of these services (in their current form).

yes. This is why it is good that we have already separate services even though we do not make full use of the possibilities yet (like running multiple instances of webui service but not liveviewhandler and websocket)

> 1. What about the Minion service? Since we rely on locking to avoid running jobs in parallel which shouldn't run in parallel I'd say there should only be one Minion service as well or multiple Minion services would share the same database. These jobs often involve a lot filesystem access under /var/lib/openqa so likely having multiple Minion services (sharing the same database) is not very beneficial if these filesystem access would then be performed from a remote host.
>
> Looks like we would really only run multiple instances of the openqa-webui.service and keep the other services on the web UI host as they are. Currently we're actually running multiple web UI processes via preforking. Maybe we could reduce the number of prefork processes then. Otherwise we would likely need to take care not to exceed the limit for PostgreSQL connections, e.g. by implementing #55262.

exactly, same as now run multiple processes but potentially spread out over different hosts and be able to dynamically alter the number and control them individually, e.g. for blue/green deployments.

> > can cause longer downtimes and make upgrades of OS more risky, e.g. when we do not have management access to VMs that might fail to reboot.
>
> But *some* host needs to provide the "entry point" and database, right? So if that host is down we would still have a downtime.

yes, but this host can be the one used for the http load-balancer which are more simple than openQA itself, have already built-in scalability features and can monitored for example by kubernetes itself.

> […]
> As long as we share the same database I would be very careful to run multiple instances of different openQA versions in parallel.

yes, careful but I would still do it :) If you like we could have a simple experimentation task first which would be "run two openQA webui instances of different versions against a single database". But I guess with just some simple practices how we introduce schema changes we should be good. If not, then just shut off all old versions as soon as the schema is newer than the currently supported one.

#5 - 2020-09-17 13:40 - mkittler

> If you like we could have a simple experimentation task first which would be "run two openQA webui instances of different versions against a single database".

We don't need an experimentation task. I can already tell you that it will sometimes work and sometimes now. If both versions use the same schema version it will work and otherwise not. The version with the newer schema will obviously migrate the database to that new version. The old version will run into database errors then. I know that too well because I often switch between different versions locally which use a different schema version.

#6 - 2020-09-23 07:38 - ilausuch

- Status changed from Workable to In Progress

- Assignee set to ilausuch

#7 - 2020-09-29 08:21 - ilausuch

Initial approach
https://github.com/os-autoinst/openQA/pull/3431

#8 - 2020-10-02 11:36 - ilausuch

Right now I am covering all the cases that has being mentioned on the PR review and also

- removing the dependence on apache
- adding a loadbalancer for the webui
- configuring webui, livehandler, websockets, gru and scheduler as independent containers

```
webui_data_1          /bin/sh -c /usr/bin/tail - ...   Up
webui_db-admin_1      entrypoint.sh docker-php-e ...   Up      0.0.0.0:8080->8080/tcp
webui_db_1            docker-entrypoint.sh postgres    Up      5432/tcp
webui_gru_1           /root/run_openqa.sh              Up      9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_haproxy_1       /docker-entrypoint.sh hapr ...   Up      0.0.0.0:9526->9526/tcp
webui_livehandler_1   /root/run_openqa.sh              Up      9526/tcp, 9527/tcp, 0.0.0.0:9528->9528/tcp, 952
9/tcp
webui_scheduler_1     /root/run_openqa.sh              Up      9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_websockets_1    /root/run_openqa.sh              Up      9526/tcp, 0.0.0.0:9527->9527/tcp, 9528/tcp, 952
9/tcp
webui_webui_1         /root/run_openqa.sh              Up      0.0.0.0:32813->9526/tcp, 9527/tcp, 9528/tcp, 95
29/tcp
webui_webui_2         /root/run_openqa.sh              Up      0.0.0.0:32814->9526/tcp, 9527/tcp, 9528/tcp, 95
29/tcp
```

#9 - 2020-10-02 14:11 - cdywan

- Related to action #43715: Update upstream dockerfiles to provide an easy to use docker image of workers added

#10 - 2020-10-02 14:12 - cdywan

- Blocks action #43712: Update upstream dockerfiles to provide an easy to use docker image of openQA-webui added

#11 - 2020-10-03 12:20 - okurz

- Blocks deleted (action #43712: Update upstream dockerfiles to provide an easy to use docker image of openQA-webui)

#12 - 2020-10-03 12:21 - okurz

- Related to action #43712: Update upstream dockerfiles to provide an easy to use docker image of openQA-webui added

#13 - 2020-10-03 12:28 - okurz

- Due date set to 2020-10-08

ilausuch wrote:

> Right now I am covering all the cases that has being mentioned on the PR review and also
>
> - removing the dependence on apache
> - adding a loadbalancer for the webui
> - configuring webui, livehandler, websockets, gru and scheduler as independent containers

Sounds great. Please keep in mind the acceptance criteria in mind and see everything that is not immediately necessary for them to be out of scope. You can note down all additional ideas in this ticket or a followup or #65271 . Also please keep https://progress.opensuse.org/projects/qa/wiki/Wiki#SLOs-service-level-objectives in mind with "aim for cycle time of individual tickets (not epics or sagas): 1h-2w". Setting due date accordingly

#14 - 2020-10-05 14:52 - ilausuch

Some changes done in the PR. If you like it I'll proceed to document this version

#15 - 2020-10-06 08:01 - ilausuch

Added the documentation to the README.md

A question, should it be retro-compatible to run without docker-compose?
In case yes, I create a new commit that allows this retro-compatibility, so if you don't use the docker-compose is expected the same behavior as before.

#16 - 2020-10-06 08:02 - ilausuch

Before to close this task I would like to check the compatibility with the workers working in this ticket #43715

#17 - 2020-10-11 12:10 - okurz

- Related to action #41600: fallback mechanism for apache, e.g. on osd added

#18 - 2020-10-23 14:17 - ilausuch

This ticket is bloqued until the mistery of https://progress.opensuse.org/issues/43715#note-22 will be solved

#19 - 2020-10-23 14:17 - ilausuch

- Status changed from In Progress to Blocked

#20 - 2020-10-23 14:48 - okurz

- Status changed from Blocked to Workable

- Assignee deleted (ilausuch)

Ok, that makes sense. We should use "blocked" only when we have another ticket reference to wait for. Let's say "workable" for someone else to pickup and "solve" that mistery :)

#21 - 2020-10-23 16:09 - okurz

- Priority changed from Normal to Low

#22 - 2020-10-26 11:51 - ilausuch

Problems solved with the worker. Now it is using nginx as LB. Preparing the auto scaling and the PR

#23 - 2020-10-26 15:41 - ilausuch

https://github.com/os-autoinst/openQA/pull/3488

This PR has the same purpose than previous one but using nginx to be able to integrate the websocket, livehandler in the path structure of the API.
This solves the communication from the workers with the api and the websockets using the same port

#24 - 2020-10-26 15:42 - ilausuch

- Status changed from Workable to Resolved

- Assignee set to ilausuch

#25 - 2020-10-26 15:42 - ilausuch

- Status changed from Resolved to In Progress

#26 - 2020-11-04 09:57 - ilausuch

- Status changed from In Progress to Resolved

Done

#27 - 2020-11-04 18:18 - okurz

- Copied to action #76990: Improve documentation for redundant/load-balancing webui deployments of openQA added

#28 - 2020-11-04 18:18 - okurz

Thank you. To provide a little bit more information: We have a proof-of-concept with the instructions so far in https://github.com/os-autoinst/openQA/blob/master/docker/README.md . What we can do as next steps is to make sure this is included in https://open.qa/docs . Not in this ticket though, included now in #76990

#29 - 2020-11-21 18:40 - okurz

- Subject changed from redundant/load-balancing webui deployments of openQA to [spike] redundant/load-balancing webui deployments of openQA

#30 - 2020-11-21 18:46 - okurz

- Parent task set to #80142

#31 - 2021-05-17 09:12 - okurz

- Due date deleted (2020-10-08)