

openQA Project - action #89731

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

coordination # 89842 (New): [epic] Scalable and streamlined docker-compose based openQA setup

containers: The deploy using docker-compose is not stable and eventually fails

2021-03-09 11:50 - ilausuch

Status:	Resolved	Start date:	2021-03-09
Priority:	Normal	Due date:	
Assignee:	ilausuch	% Done:	0%
Category:	Concrete Bugs	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			

Description

Motivation

The command 'docker-compose up' is executed without errors in normal circumstances, but sometimes some of the containers fail later after the docker-compose has finished.

```
$ docker-compose up -d
Creating webui_db_1      ... done
Creating webui_nginx_1  ... done
Creating webui_data_1   ... done
Creating webui_scheduler_1 ... done
Creating webui_webui_1  ... done
Creating webui_webui_2  ... done
Creating webui_gru_1    ... done
Creating webui_websockets_1 ... done
Creating webui_livehandler_1 ... done
$ echo $?
0
```

```
docker-compose ps
```

Name	Command	State	Ports
webui_data_1	/bin/sh -c /usr/bin/tail - ...	Up	
webui_db_1	docker-entrypoint.sh postgres	Up	5432/tcp
webui_gru_1	/root/run_openqa.sh	Up	443/tcp, 80/tcp, 9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_livehandler_1	/root/run_openqa.sh	Up	443/tcp, 80/tcp, 9526/tcp, 9527/tcp, 0.0.0.0:9528->9528/tcp, 9529/tcp
webui_nginx_1	/entrypoint.sh	Up	0.0.0.0:9526->9526/tcp
webui_scheduler_1	/root/run_openqa.sh	Exit 255	
webui_websockets_1	/root/run_openqa.sh	Up	443/tcp, 80/tcp, 9526/tcp, 0.0.0.0:9527->9527/tcp, 9528/tcp, 9529/tcp
webui_webui_1	/root/run_openqa.sh	Up	443/tcp, 80/tcp, 0.0.0.0:32789->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_webui_2	/root/run_openqa.sh	Up	443/tcp, 80/tcp, 0.0.0.0:32790->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp

The errors in schedulers are:

```
scheduler_1 | failed to run SQL in /usr/share/openqa/script/../../dbicdh/PostgreSQL/deploy/90/001-auto-__VERSION.sql: DBIx::Class::DeploymentHandler::DeployMethod::SQL::Translator::try {...} (): D
```

```
BI Exception: DBD::Pg::db do failed: ERROR: duplicate key value violates unique constraint "pg_type_tname_nsp_index"
scheduler_1 | DETAIL: Key (typename, typnamespace)=(dbix_class_deploymenthandler_versions_id_seq, 2200) already exists. at inline delegation in DBIx::Class::DeploymentHandler for deploy_method->deploy (attribute declared in /usr/lib/perl5/vendor_perl/5.26.1/DBIx/Class/DeploymentHandler/WithApplicatorDumple.pm at line 51) line 18
scheduler_1 | (running line 'CREATE TABLE dbix_class_deploymenthandler_versions ( id serial NOT NULL, version character varying(50) NOT NULL, ddl text, upgrade_sql text, PRIMARY KEY (id), CONSTRAINT dbix_class_deploymenthandler_versions_version UNIQUE (version) )') at /usr/lib/perl5/vendor_perl/5.26.1/DBIx/Class/DeploymentHandler/DeployMethod/SQL/Translator.pm line 263.
scheduler_1 | DBIx::Class::Storage::TxnScopeGuard::DESTROY(): A DBIx::Class::Storage::TxnScopeGuard went out of scope without explicit commit or error. Rolling back. at /usr/share/openqa/script/openqa-scheduler line 0
scheduler_1 | DBIx::Class::Storage::TxnScopeGuard::DESTROY(): A DBIx::Class::Storage::TxnScopeGuard went out of scope without explicit commit or error. Rolling back. at /usr/share/openqa/script/openqa-scheduler line 0
```

The problem is that every container that uses openqa_webui image (webui_webui, webui_websockets, webui_scheduler, webui_livehandler) try to initialize the DB tables. And as all the containers are initialized at the same time surges conflicts.

Acceptance Criteria

- **AC 1:** All the containers remain up after execute docker-compose up *~~AC 2: Expand the docker-compose CI test to include this ease~~

Suggestions

- Use dependencies (depends_on) based on health-checks to sort the startup of all the containers.
- Check current solution on <https://github.com/os-autoinst/openQA/pull/3755>

Related issues:

Related to openQA Project - action #91046: CI: "webui-docker-compose" seems t...	Resolved	2021-04-13
Blocked by openQA Project - action #89719: docker-compose up fails on master	Resolved	2021-03-09
Blocked by openQA Project - action #89722: Need automatic check for docker-co...	Resolved	2021-03-09

History

#1 - 2021-03-09 11:52 - ilausuch

- Description updated

#2 - 2021-03-09 12:11 - cdywan

- Blocked by action #89719: docker-compose up fails on master added

#3 - 2021-03-09 12:12 - cdywan

I think logically this will come after [#89719](#) hence marking this as *Blocked*, since the AC assume an existing compose test that can be extended

#4 - 2021-03-09 12:18 - cdywan

- Blocks action #76978: How to run an openQA test in 5 minutes added

#5 - 2021-03-09 12:52 - cdywan

cdywan wrote:

I think logically this will come after [#89719](#) hence marking this as *Blocked*, since the AC assume an existing compose test that can be extended

Correction. Since [okurz](#) filed a separate issue this can actually be blocked by [#89722](#)

#6 - 2021-03-09 12:52 - cdywan

- Blocks deleted (action #76978: How to run an openQA test in 5 minutes)

#7 - 2021-03-09 12:53 - cdywan

- Blocked by action #89722: Need automatic check for docker-compose added

#8 - 2021-03-09 15:47 - ilausuch

I checked that the list of running containers could be checked with
docker-compose ps --service --filter status=running

Unfortunately I checked using status=exited but only works on docker but not on docker-compose (At least in the version 1.27.4)

An other option is to use docker-compose ps | grep Exit but this has the problem that none of the container have a name containing Exit

#9 - 2021-03-09 16:18 - cdywan

ilausuch wrote:

I checked that the list of running containers could be checked with
docker-compose ps --service --filter status=running

Unfortunately I checked using status=exited but only works on docker but not on docker-compose (At least in the version 1.27.4)

An other option is to use docker-compose ps | grep Exit but this has the problem that none of the container have a name containing Exit

I would approach it from the mojo app first before looking at the containers. If e.g. database access is not locked that would be a problem for other use cases, too.

#10 - 2021-03-12 09:00 - okurz

- Target version set to future

#11 - 2021-03-12 09:31 - cdywan

Target version set to future

From the description it seemed like a pretty serious bug to me. Did you confirm that it's not?

#12 - 2021-03-12 10:41 - okurz

No, I did not. We did the exception to originally accept the docker-compose.yaml without automatic tests. So we just learned again what we did know anyway: Every code contribution needs an automatic test. Gladly the docker-compose approach is not prominently mentioned in documentation yet.

So [#89722](#) first

#13 - 2021-03-12 10:48 - cdywan

okurz wrote:

No, I did not. We did the exception to originally accept the docker-compose.yaml without automatic tests. So we just learned again what we did know anyway: Every code contribution needs an automatic test. Gladly the docker-compose approach is not prominently mentioned in documentation yet.

So [#89722](#) first

Sure. I just thought that's conveyed by the blocked status and not necessary to handle manually.

#14 - 2021-03-12 11:14 - okurz

- Parent task set to [#89842](#)

#15 - 2021-03-31 11:25 - ilausuch

I ran a test that fails

https://github.com/os-autoinst/openQA/pull/3818/checks?check_run_id=2236273686

I only did a sleep before check that all the containers are UP to grant time to the failure.

#16 - 2021-04-06 11:39 - ilausuch

- Status changed from New to In Progress

#17 - 2021-04-06 11:39 - ilausuch

There is a draft PR with some approaches

<https://github.com/os-autoinst/openQA/pull/3821>

The problem that I have detected is that all the webui containers try to create and initialize the DB at the same time, and since all the webui replicas are launched at the same time by the docker-compose and we have more than one, it generates the problem
My approach has been to create an initial webui and sort the dependencies so that the webui cluster is only initialized after initializing this first webui. In addition, I base the dependencies on the health checks ensuring that it does not go to the next phase unless the previous one is complete.

#18 - 2021-04-07 06:23 - cdywan

- Target version changed from future to Ready

I gather this is Ready then, after confirming with [okurz](#) and [ilausuch](#) that there was an internal conversation about this ticket not reflected here.

#19 - 2021-04-07 15:46 - ilausuch

Why it takes >30min?

I think the process spend a lot of time building the images. And in empirical test in my local machine, the installation of packages is the most tedious part. But I think this should measured to be sure.

One idea:

We need to build these images because maybe the PR that is launching the test do some changes that should be tested. How ever, we could base these images in a previous build image where all the packages are already installed. This will accelerate the process of testing. At least it is something that I have already tried in my experience

#20 - 2021-04-07 16:16 - cdywan

ilausuch wrote:

Why it takes >30min?

Was that comment meant for [#90767](#) ?

#21 - 2021-04-13 09:23 - ilausuch

- Related to action #91046: CI: "webui-docker-compose" seems that eventually fails again added

#22 - 2021-04-19 09:50 - okurz

- Due date set to 2021-04-21

- Assignee set to ilausuch

[ilausuch](#) I assume you should be assignee due to [#89731#note-16](#) ?

#23 - 2021-04-20 10:44 - ilausuch

We have here a new situation under investigation

https://github.com/os-autoinst/openQA/pull/3850/checks?check_run_id=2388942783

Webui_1 is unhealthy

Name	Command	State	Port
webui_db_1	docker-entrypoint.sh postgres	Up (healthy)	5432/tcp
webui_webui_1	/root/run_openqa.sh	Up (unhealthy)	443/tcp, 80/tcp, 0.0.0.0:49155->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_webui_2	/root/run_openqa.sh	Up (healthy)	443/tcp, 80/tcp, 0.0.0.0:49154->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_webui_db_init_1	sh -c chmod -R a+rwX /data ...	Up (healthy)	443/tcp, 80/tcp, 0.0.0.0:49153->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp

#24 - 2021-04-20 14:37 - ilausuch

Solved with <https://github.com/os-autoinst/openQA/pull/3850>

#25 - 2021-04-23 12:05 - cdywan

ilausuch wrote:

Solved with <https://github.com/os-autoinst/openQA/pull/3850>

So is [#91377](#) a duplicate of this? Or vice versa? Is the same problem being solved here?

#26 - 2021-04-26 08:06 - ilausuch

No, [#91377](#) was for the static check and this if for the docker-compose test

#27 - 2021-04-26 11:48 - ilausuch

The AC1 is solved with <https://github.com/os-autoinst/openQA/pull/3821>

#28 - 2021-04-27 05:32 - ilausuch

About the AC2 all the container should be with a healthy status.

But we can see here that nginx (that is the last container in the chain) needs some time to stand-up.

https://github.com/os-autoinst/openQA/pull/3864/checks?check_run_id=2444880438

Name	Ports	Command	State
webui_db_1		docker-entrypoint.sh postgres	Up (healthy) 5432/tcp
webui_gru_1		sh -c /root/run_openqa.sh ...	Up (healthy) 443/tcp, 80/tcp, 9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_livehandler_1		/root/run_openqa.sh	Up (healthy) 443/tcp, 80/tcp, 9526/tcp, 9527/tcp, 0.0.0.0:9528->9528/tcp, :::9528->9528/tcp, 9529/tcp
webui_nginx_1		/entrypoint.sh	Up (health: starting) 0.0.0.0:9526->9526/tcp, :::9526->9526/tcp
webui_scheduler_1		/root/run_openqa.sh	Up (healthy) 443/tcp, 80/tcp, 9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_websockets_1		/root/run_openqa.sh	Up (healthy) 443/tcp, 80/tcp, 9526/tcp, 0.0.0.0:9527->9527/tcp, :::9527->9527/tcp, 9528/tcp, 9529/tcp
webui_webui_1		/root/run_openqa.sh	Up (healthy) 443/tcp, 80/tcp, 0.0.0.0:49155->9526/tcp, :::49155->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_webui_2		/root/run_openqa.sh	Up (healthy) 443/tcp, 80/tcp, 0.0.0.0:49154->9526/tcp, :::49154->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp
webui_webui_db_init_1		sh -c chmod -R a+rwX /data ...	Up (healthy) 443/tcp, 80/tcp, 0.0.0.0:49153->9526/tcp, :::49153->9526/tcp, 9527/tcp, 9528/tcp, 9529/tcp

I am thinking in two possible solutions

- Give some time to nginx to start with an active waiting until healthy or unhealthy that are "final statuses"
- Create a dummy container dependent of nginx to ensure that nginx has started with healthy

#29 - 2021-04-27 09:37 - ilausuch

The AC2 has moved to [#91815](#). As far as AC1 is covered (<https://progress.opensuse.org/issues/89731#note-27>) this ticket can be considered resolved. do you Agree?

#30 - 2021-04-27 09:44 - ilausuch

- Description updated
- Status changed from In Progress to Resolved

#31 - 2021-05-17 09:14 - okurz

- Due date deleted (2021-04-21)