

openQA Project - action #73309

every time a direct dependency is updated in Factory our CI jobs fail until the package is updated

2020-10-13 21:41 - okurz

Status:	Workable	Start date:	2020-10-13
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	future		
Difficulty:			

Description

Observation

<https://github.com/os-autoinst/openQA/pull/3463> is a very simple change. The CI jobs fail immediately on circleCI in the "cache" step with:

```
+ sudo zypper -n install --download-only aspell-0.60.6.1...ShellCheck-0.6.0
Loading repository data...
Reading installed packages...
Package 'perl-Mojolicious-8.61' not found.
```

because the package perl-Mojolicious-8.61 does not exist anymore in this version in devel:openQA, devel:openQA:Leap:15.1 and openSUSE:Factory as it is already at 8.62 so the old files vanished. Retriggering the PR as well as rebasing the PR would not help because only any next nightly job would update the version specific dependency list.

Problem

The problem was introduced when we moved to circleCI together with a "new" approach to hardcode versions of packages so that pull requests would not suddenly fail if any updated dependency would by coincidence appear the first time. However it feels like now we have more or less the same problem, less subtle failures maybe but even more likely than what we wanted to avoid.

Acceptance criteria

- **AC1:** Pull requests do not fail tests just because suddenly a package has a new version

Suggestions

- Research how other projects cope with this
- Try to find an approach to make our dependencies we test consistent but available. How about having *all* dependencies pre-installed in a container and test based on that?

Related issues:

Related to openQA Project - action #57050: Turn off travis	Resolved	2019-09-30
Related to openQA Project - action #55478: Evaluate circleci for openQA	Resolved	2019-08-14
Related to openQA Project - action #56522: Create cron job which will create ...	Resolved	2019-08-14
Related to openQA Project - action #40580: Run openQA travis tests against up...	Resolved	2018-09-04
Related to openQA Project - action #53546: Easier dependencies handling for p...	New	2019-06-27
Related to openQA Project - action #88837: Separate out style checks into a d...	New	2021-02-19

History

#1 - 2020-10-13 21:42 - okurz

Can anybody remind me why we don't pre-install all dependencies into a container that we use for tests? Didn't we want to use openQA-devel for that but still call zypper on top in case anyone specifies new dependencies in a PR?

#2 - 2020-10-13 21:43 - okurz

- Related to action #57050: Turn off travis added

#3 - 2020-10-13 21:43 - okurz

- Related to action #55478: Evaluate circleci for openQA added

#4 - 2020-10-13 21:43 - okurz

- Related to action #56522: Create cron job which will create pull requests with list of current openQA-devel dependencies with version added

#5 - 2020-10-13 21:43 - okurz

- Related to action #40580: Run openQA travis tests against upgraded CPAN dependencies added

#6 - 2020-10-13 21:43 - okurz

- Related to action #53546: Easier dependencies handling for packages, e.g. reduce duplication of build requirements in spec, documentation, Dockerfile added

#7 - 2020-10-14 09:13 - tinita

We could use the same approach as we have in os-autoinst now.

It uses a container with preinstalled dependencies and additionally installs all of them to get any updates.

Downsides:

- This doesn't handle *removed* dependencies yet (but this should be possible to implement).
- We get new dependencies as soon as they are available, so we don't know what module versions we get in advance.

#8 - 2020-10-14 11:52 - andriinikitin

It is not that straightforward for CI: cache step will try to download packages only if cache is not there.

So, the flow only fails when: the fork never used ci-packages.txt since it was created by openqabot and dependencies have changed since that. (If only CircleCI would allow to share cache between upstream and fork, that would not happen)

But again, if dependencies have changed - the openqabot will create a new PR to reflect that the same day (and test new layout accordingly).

So it is relatively little chance for the problem to happen, at least not for active developer who would create cache in CI when it is still relevant

And again, if your fork has problem with ci-packages.txt - you can just do what openqabot does: update ci-packages.txt to relevant state in your fork: just run

```
bash .circleci/build_dependencies.sh (needs docker) and then add updated ci-packages.txt to your branch.
```

As an alternative approach one can reuse existing cache in upstream by pushing your branch there, in case if your fork has problems with dependencies. Or even use openqabot's fork for that (we should try if that will work).

We get new dependencies as soon as they are available, so we don't know what module versions we get in advance.

In my understanding it is not what we want, because current CircleCI workflow guarantees that every time you use 'known' ci-packages.txt : you are secured about eventual problems which dependencies can bring. It needs some additional effort to guarantee the same with container. Also if the container gets problematic dependency - every commit will have problems until container is fixed.

So, the ticket needs another Acceptance Criteria:

AC2: Every commit in the same branch is tested against exactly the same dependencies, unless the change of dependencies is explicitly stated somewhere (e.g. in ci-packages.txt)

#9 - 2020-10-14 13:51 - okurz

andriinikitin wrote:

It is not that straightforward for CI: cache step will try to download packages only if cache is not there.

So, the flow only fails when: the fork never used ci-packages.txt since it was created by openqabot and dependencies have changed since that. (If only CircleCI would allow to share cache between upstream and fork, that would not happen)

But again, if dependencies have changed - the openqabot will create a new PR to reflect that the same day (and test new layout accordingly).

So it is relatively little chance for the problem to happen, at least not for active developer who would create cache in CI when it is still relevant

This is not what I observed. It seems to happen for basically everyone including the most active developers. I have the suspicion that the cache is never used. But also the cache does not include all dependencies but only the basics. The problem is about the further dependencies.

[...]

In my understanding it is not what we want, because current CircleCI workflow guarantees that every time you use 'known' ci-packages.txt : you are secured about eventual problems which dependencies can bring. It needs some additional effort to guarantee the same with container. Also if the container gets problematic dependency - every commit will have problems until container is fixed.

I think you can easily reproduce the problem by creating a branch from an older commit with an older ci-packages.txt

So, the ticket needs another Acceptance Criteria:

AC2: Every commit in the same branch is tested against exactly the same dependencies, unless the change of dependencies is explicitly stated somewhere (e.g. in ci-packages.txt)

If we can combine that with AC1 I agree but I am not sure if we can actually do that or if even by theory they both conflict

#10 - 2020-10-14 14:12 - andriinikitin

okurz wrote:

This is not what I observed. It seems to happen for basically everyone including the most active developers. I have the suspicion that the cache is never used.

Two reasons I can imagine:

1. dependencies change too often, so ci-packages.txt becomes obsolete before fork can use it.
2. the bot fails to update ci-packages.txt timely (I did see nightly job failed often regularly).

But also the cache does not include all dependencies but only the basics. The problem is about the further dependencies.

The cache for sure stores all indirect dependencies as well (excluding those that come in ci base container:

<https://github.com/os-autoinst/openQA/blob/master/docker/devel.openQA:ci/base/Dockerfile>)

I think you can easily reproduce the problem by creating a branch from an older commit with an older ci-packages.txt

The CI cache is stored for 30 days, so it is kind of expected that we cannot test in exactly the same layout. But again - I think we want to control all dependencies, thus explicitly regenerating ci-packages.txt is acceptable payoff.

So, the ticket needs another Acceptance Criteria:

AC2: Every commit in the same branch is tested against exactly the same dependencies, unless the change of dependencies is explicitly stated somewhere (e.g. in ci-packages.txt)

If we can combine that with AC1 I agree but I am not sure if we can actually do that or if even by theory they both conflict

I clearly remember times when Travis-CI was red and nobody could do anything for several days when container was broken (e.g. when new tidy was released). So implementing AC1 doesn't fix anything, just shifts the problem into worse direction.

I think being fluent with manually controlling ci-packages.txt is good pay off for benefits that current CI setup can offer.

#11 - 2020-10-29 21:24 - okurz

[andriinikitin](#) sorry if you feel like you need to repeat yourself :) In <https://github.com/os-autoinst/openQA/pull/3493> I see that again the "cache" job fails: <https://app.circleci.com/pipelines/github/os-autoinst/openQA/4631/workflows/6ea9ec49-b4bb-4d67-821d-137d874c8c43/jobs/44244> shows:

```
+ sudo zypper -n install --download-only aspell-0.60.8 ... python3-yamllint-1.22.1 ShellCheck-0.7.1
Loading repository data...
Reading installed packages...
Package 'perl-Minion-10.13' not found.
Package 'perl-Mojo-Pg-4.20' not found.
```

which is because these versions "perl-Minion-10.13" and the other do not exist anymore in the repositories that zypper added. But if I understand you correctly this only happened because I did not create a pull request from my fork for a long time or never? That can't be true because I would consider myself active enough that this should normally never happen :) You are stating "The CI cache is stored for 30 days". Can we see that in practice in any circleCI job where is shown to be working?

EDIT: After I now rebased my branch I ended up in

<https://app.circleci.com/pipelines/github/os-autoinst/openQA/4674/workflows/b9c96d8b-8b57-4c17-879c-15f037a023f8/jobs/44612> with

```
Package 'perl-Cpanel-JSON-XS-4.24' not found.
```

because likely a newer of that package is in Tumbleweed now and the old version was removed. Sure, I can update dependencies but the point is that with just my trivial code changes the CI jobs fail and only after they fail I would need to try to fix them now which costs time.

#12 - 2020-10-29 22:04 - okurz

- Target version changed from Ready to future

#13 - 2021-02-19 16:01 - cdywan

- Related to action #88837: Separate out style checks into a dedicated environment added