

openQA Project - action #18000

coordination # 13812 (New): [epic][dashboard] openQA Dashboard ideas

[dashboard] Better integration of status from external jobs and testing systems

2017-03-24 14:11 - mkittler

Status: Workable	Start date: 2017-03-24
Priority: Low	Due date:
Assignee:	% Done: 0%
Category: Feature requests	Estimated time: 0.00 hour
Target version: future	
Difficulty:	

Description

User story

As a tester I want to use an easy way to publish my test results on an openQA server to use openQA as a dashboard for test results

acceptance criteria

- **AC1:** There is a way to fill the whole jobs details page without the need to use os-autoinst, e.g. console client to fill "Details" and "Logs & Assets"
- **AC2:** The whole workflow is documented

tasks

- Come up with a way how the console tool should upload the missing information, e.g. how the command line parameters are called, e.g. "openqa-client upload_screenshot ..."
- document the current API used internally by the openQA backend to uploads results, e.g. mandatory variables.
- optional: some refactoring and improvements can be done to ease using that API by other tools, too.

further details

goal: To use the openQA dashboard as the main reporting platform also for other tools we have in our infrastructure (at SUSE).

There are already test runners feeding back the information e.g. in junit format. Also, using the client script jobs can be started and their result set along with comments, e.g. for tags. Example:

```
job_id=$(client --host http://localhost jobs post TEST=foo_external _GROUP="openSUSE Tumbleweed AA  
rch64" BUILD=1 | sed -n 's/^{ id => \([0-9]\+\) }$/\1/p')
```

```
client --host http://localhost jobs/${job_id}/set_done post result=passed  
client --host http://localhost jobs/${job_id}/comments post text=bsc#1234
```

Related issues:

Related to qe-yast - coordination #42890: [functional][y][epic] Established w...	New	2018-10-24
--	------------	-------------------

History

#1 - 2017-03-27 16:00 - RBrownSUSE

- Subject changed from [tools] Better integration of status from external jobs and testing systems to [tools][dashboard] Better integration of status from external jobs and testing systems

#2 - 2017-05-16 12:54 - krauselukas

- Description updated

- Category set to Feature requests

#3 - 2017-11-23 12:26 - coolo

- Status changed from New to Rejected

To upload status from external tools use the worker api and not client api calls. As such set_done should be forbidden from any other than workers.

We're currently developing a federation worker bridge, which could be reused for such tools - if we need them in practise.

#4 - 2017-11-28 09:19 - mkittler

As far as I remember the request for this feature didn't say anything about the particular API to be used.

It was about providing *some* way to add information to openQA like os-autoinst does. Obviously this is already implemented from the openQA-side, otherwise os-autoinst could not do its job.

So this issue was about *documenting a workflow*, eg. how to use the worker API and maybe provide some helper scripts to ease the task. So this is actually not about modifying openQA, it is about providing help to use it. (Richard likely tagged it with [dashboard] because the goal is to display external jobs results in the dashboard.)

We're currently developing a federation worker bridge, which could be reused for such tools

Not sure what this 'federation worker bridge' would be, but it might help.

if we need them in practise

Don't know whether this issue is still relevant, too. Note that it was not my idea, I've just wrote it down.

#5 - 2017-11-29 16:06 - okurz

- Status changed from Rejected to In Progress

mkittler wrote:

[...]

if we need them in practise

Don't know whether this issue is still relevant, too. Note that it was not my idea, I've just wrote it down.

Yes, we still need this "in practice". Agreeing with mkittler here. He has added the request also based on my suggestions and because I asked him to so let's please handle this as a feature request that is still open. I don't see any information missing regarding the request. The request is still a valid one also by e.g. "QA SLE kernel", e.g. see <https://github.com/os-autoinst/openQA/pull/1492> . I think I know what "federation" is but I don't see that as related.

#7 - 2017-12-12 13:37 - metan

Let me just reference action #27726 [kernel] NVMe Over Fabrics Unit Test Framework here. We are asked to integrate nose2 based testsuite into the openQA. The obvious question is how export the test results so that we have something meaningful in the openQA result page. I looked into nose2 and it looks like you can implement custom result formatter class in python or use existing one that produces junit.

So the question is how will we handle that? For the action #25680 the best solution seems to be to implement an openQA API function to report the test result since the testcases are executed in the openQA test Perl code since we implemented the test runner from the scratch there. But in this case I doubt that we can call openQA API functions from the python class without significant overhead so file based result export sounds like reasonable solution.

Any ideas how to handle nose2 based tests?

#8 - 2017-12-12 13:49 - okurz

Who asked you to "integrate nose2 based testsuite into the openQA"?

I see the following options:

1. Execute the "unit test framework" (it's calling itself this way) in OBS/IBS (preferred)
2. just execute nosetests --with-xunit within the SUT in the test framework directory and make sure in openQA the xunit format is properly parsed and visualized

#9 - 2017-12-12 13:52 - coolo

For parsing/rendering the xunit, we work on <https://progress.opensuse.org/issues/16076> (misleading title, trust me :) as part of the current sprint

#10 - 2017-12-12 14:10 - metan

[okurz](#) Well it's actually kernel testsuite that happens to use nose2 so the request came from kernel labs team.

#11 - 2017-12-12 14:21 - EDiGiacinto

i will report here for reference what i've added in <https://github.com/os-autoinst/openQA/pull/1492>.

Since mostly the issues are pointing to the same direction ('an uniform' way to treat test results by other frameworks in openQA) to ease out this situation, i'm working in a generic parsing solution that allows at least to include external test in an (almost) uniform way, translating them in an internal test representation that can be serialized and deserialized - with the possibility to extend it with different formats and also to map them as openQA test modules run if needed.

That's my branch https://github.com/mudler/openQA/tree/test_parser, currently the parser is able to read/write/translate JUnit - will extend it to the JSON format proposed by @richiejp and XUnit - so we can use it to serialize/deserialize external tests results inside the openQA DB.

#12 - 2018-01-19 14:50 - okurz

<https://github.com/okurz/scripts/blob/master/openqa-client-record-manual-test> should show what I mean with an easy way to record results, e.g. from manual testing, with one go.

#13 - 2019-02-22 11:51 - okurz

- *Related to coordination #42890: [functional][y][epic] Established workflow how system level test failures found in openQA of QA validation tests can be moved to earlier testing, e.g. staging, OBS devel projects, PR, commit added*

#14 - 2019-07-30 09:18 - okurz

- *Status changed from In Progress to Workable*

by now not really "In Progress", back to "Workable".

#15 - 2020-04-03 15:31 - okurz

- *Subject changed from [tools][dashboard] Better integration of status from external jobs and testing systems to [dashboard] Better integration of status from external jobs and testing systems*

- *Priority changed from Normal to Low*

trying to set prio of parent to "Low" implicitly.

#16 - 2020-07-28 11:30 - okurz

- *Target version set to future*