

## openQA Project - action #20246

### [gsoc] Investigate/implement container-based backend

2017-07-04 09:39 - EDiGiacinto

<b>Status:</b>	Rejected	<b>Start date:</b>	2017-07-04
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	okurz	<b>% Done:</b>	0%
<b>Category:</b>	Feature requests	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	future		
<b>Difficulty:</b>			
<b>Description</b>			
As spoken briefly with Richard, i couldn't find an already open issue regarding containers support in the backend, so opening a new one.			
<b>User story</b>			
As a tester, i want to be able to test containers and validating images before going further down in the pipelines. (e.g. testing images before being used for deploying container-based services in multi-machine environments)			
<b>acceptance criteria</b>			
<ul style="list-style-type: none"><li>• <b>AC1:transparent</b> implementation to support different types of containerization technologies ( docker, rkt, lxc .. )</li><li>• <b>AC2:</b> do not have to increase tweakings or options to overwhelm the tester</li><li>• <b>AC3:</b> try to provide maximum compatibility on features that we use already in VM-based backends</li><li>• ...</li></ul>			
<b>tasks</b>			
<ul style="list-style-type: none"><li>• Investigate required changes to the codebase and look into cpan if there are modules that we could use</li><li>• Provide basic support to handle containers operations in the backend side</li><li>• Add support for the docker backend (as we use it in some of our products)</li><li>• ...</li></ul>			
<b>further details</b>			
Investigate if makes sense implement a backend for containers. In case it is, we should discuss also what container-based backend(s) we would like to add support to.			
A backend capable of directly testing containers could be useful since we provide container images, and might add business value as well ( as CaaS is using docker ).			
<b>Related issues:</b>			
Related to openQA Project - action #30074: Use container as a product under test		<b>Rejected</b>	<b>2018-01-09</b>
Related to openQA Tests - action #12804: [qe-core][functional][sles][opensuse...		<b>Workable</b>	<b>2016-07-21</b>

#### History

##### #1 - 2017-07-04 10:50 - coolo

testing containers within reference vms is good enough for the moment - and we have other requirements that are not possible at all. So I'm inclined to set this to future milestone

##### #2 - 2017-11-22 08:36 - coolo

- Subject changed from [tools]Investigate/implement container-based backend to Investigate/implement container-based backend
- Target version set to future

##### #3 - 2018-01-10 08:34 - EDiGiacinto

- Related to action #30074: Use container as a product under test added

**#4 - 2018-01-19 16:08 - EDiGiacinto**

- Subject changed from *Investigate/implement container-based backend* to *[gsoc] Investigate/implement container-based backend*

Add [gsoc] tag, see: <https://github.com/openSUSE/mentoring/issues/92>

**#5 - 2018-01-24 16:48 - szarate**

Talking to panos, he mentioned this (Which seems to be using systemd-nspawn) <https://github.com/drpaneas/egkatasistasis>

**#6 - 2018-06-15 19:08 - okurz**

- Target version changed from *future* to *future*

**#8 - 2018-10-18 13:15 - okurz**

- Related to action #12804: *[qe-core][functional][sles][opensuse][installation] Do more easy to debug tests by using chroot installations ... or containers* :) added

**#9 - 2019-06-20 15:30 - okurz**

- Category changed from *132* to *Feature requests*

**#10 - 2020-02-04 08:31 - okurz**

opened again for gsoc, see <https://github.com/openSUSE/mentoring/issues/120>

**#11 - 2020-10-29 20:36 - okurz**

- Status changed from *New* to *Rejected*

- Assignee set to *okurz*

The issue for GSOC is still open. For us in this ticket I assess that we do not need to keep this reference. The current approach to test container images within openQA tests is still as mentioned in <https://progress.opensuse.org/issues/20246#note-1> within the scope of a surrounding OS running the container images. This has the benefit of keeping it visible to test reviewers what the actual surrounding environment is which can have quite some impact on the executed container instances. Just for comparison our s390x z/VM as well as s390x kvm backend is trying to "abstract away" the surrounding layers including an additional X server, vncviewer and such. As this layer is not without problems this is making it hard for test reviewers to understand what is going on so also not the best approach.