

openQA Project - action #81118

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

coordination # 43706 (Blocked): [epic] Generate "download&use" docker image of openQA for SUSE QA

automatic container tests for os-autoinst

2020-12-16 15:32 - okurz

Status:	Resolved	Start date:	2020-12-16
Priority:	Normal	Due date:	
Assignee:	okurz	% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			
Description			

History

#1 - 2020-12-16 19:43 - okurz

- Tracker changed from coordination to action

cdywan, ilausuch and okurz discussed the topic of container testing again today in daily meeting 2020-12-16. As we did not succeed with a plain test script within openQA context I tried to explore alternatives and research about "state of art" myself.

Putting openQA itself into the game is a nice idea but maybe we can come up with something simpler. I am trying in os-autoinst where I added Dockerfile's recently but the concept can be the same. I created a file tools/test_containers with content:

```
#!/bin/bash
```

```
set -e pipefail
```

```
cre="${cre:-"docker"}"
for i in container/*/Dockerfile*; do
    tag=${i,,}
    $cre build -t $tag -f $i $(dirname $i)/
    $cre run --rm -it $tag --help
done
```

this should try to build and run every container we have in our corresponding repo each. We can set the variable \$cre (CRE For "container runtime environment"), e.g. to select docker or podman. Following man podman for rootless I did locally

```
sudo usermod --add-subuids 10000-75535 $USER
sudo usermod --add-subgids 10000-75535 $USER
```

but still get ERRO[0006] Error while applying layer: ApplyLayer exit status 1 stdout: stderr: there might not be enough IDs available in the namespace (requested 0:15 for /etc/shadow): lchown /etc/shadow: invalid argument. Likely the same problem ilausuch had. Based on docker I can succeed in both a local env as well as github actions.

I created <https://github.com/os-autoinst/os-autoinst/pull/1602> exploring the concept.

A local run of tools/test_containers runs fine but of course takes some time to download and install a similar set of packages repeatedly. Also we have to keep in mind that our Dockerfiles are based on published openSUSE packages so we do not test the state of the source repo except for the Dockerfile based container image construction itself.

I am thinking that for now for PR checks I would just include static checks of Dockerfiles in both os-autoinst as well as openQA. The container build+run tests I would only call in master due to the increased runtime. Maybe we can reuse the zypper package cache though.

#2 - 2020-12-17 04:07 - openqa_review

- Due date set to 2020-12-31

Setting due date based on mean cycle time of SUSE QE Tools

#3 - 2020-12-17 20:21 - okurz

- Status changed from In Progress to Feedback

#4 - 2021-01-05 13:11 - cdywan

- Due date changed from 2020-12-31 to 2021-01-08

Bumping the Due date for now. Maybe we can discuss what we have, and what we still want later this week.

#5 - 2021-01-11 08:37 - okurz

- Status changed from Feedback to Resolved

<https://github.com/os-autoinst/os-autoinst/actions?query=branch%3Amaster> shows successful runs, e.g.

https://github.com/os-autoinst/os-autoinst/runs/1656678554?check_suite_focus=true showing the "ci_extended" tests with building and executing the container images

#6 - 2021-03-18 13:18 - okurz

- Due date deleted (2021-01-08)