openQA Project - coordination #40199

[EPIC] Better rollback capabilities of (worker) deployments

2018-08-23 19:30 - okurz

Status:	Resolved	**Start date:**	2018-08-23
Priority:	Normal	**Due date:**	
Assignee:	okurz	**% Done:**	0%
Category:	Feature requests	**Estimated time:**	0.00 hour
Target version:	future		
Difficulty:			

Description

As an outcome of #39743 we learned Complete deployment rollbacks for the whole infrastructure would be nice (including openQA packages, database and test settings, system packages on both web UI as well as workers) but there will always be factors which are changing outside our control

Related issues:

Related to openQA Project - action #39743: [o3][tools] o3 unusable, often res...	**Resolved**	**2018-08-15**

History

#1 - 2018-08-23 19:33 - okurz

- Related to action #39743: [o3][tools] o3 unusable, often responds with 504 Gateway Time-out added

#2 - 2018-09-03 12:08 - coolo

I have my doubts that this is a reasonable request. Wishful thinking might lead to this, but IMO you need quite a deployment team to roll something like this. And on top of that: I don't think the deployment strategy is part of the 'openQA Project'.

#3 - 2018-09-17 07:19 - coolo

- Subject changed from [tools] Better rollback capabilities of deployments to [EIPC] Better rollback capabilities of (worker) deployments

- Target version set to future

For the webui I have no good idea - but for the workers we could complement salt with workers reinstalling a defined state on boot. This is still a huge task - and work intensive to maintain, so I'm not really sure we should invest there.

#4 - 2018-09-17 14:47 - okurz

- Subject changed from [EIPC] Better rollback capabilities of (worker) deployments to [EPIC] Better rollback capabilities of (worker) deployments

I guess you meant "EPIC" instead of "EIPC" ;) You are loosing some part of the original idea when you restrict it with "(worker)" and not cover the web UI part anymore. I am with you that this is no easy "let's hack some perl" task but still I see it as feasible. And isn't this basically also a business case we sell to customers? At least on feasible – albeit maybe not the best – approach to reach the goal of the (original) ticket description would be:

- Use btrfs with snapshots on / for each machine (done for workers, missing for webui)
- Only ever upgrade the webui together with a full database dump saved just before the upgrade (script or salt should work)
- Train dry-runs with all involved admins of the "worst case scenarios" to have them less scared and reduce the recovery time in case of emergencies
- Optional: Save RPM files used for installation on both webui + worker elsewhere to be able to go back to or automatic maintenance requests for tested packages based on openQA-in-openQA which makes sure that older versions of package are saved "automatically" but probably the openQA updates are too heavy for the maintenance workflow

#5 - 2019-06-20 17:18 - okurz

- Category changed from 168 to Feature requests

#6 - 2019-08-20 20:34 - okurz

so for o3 what works quite well is to have transactional server worker hosts and for the o3 webui host keep packages from devel:openQA repos, a simple keeppackages=1 in the .repo files. We commonly save a database dump when we update the webui host so that part is also covered. And also we have automation for the complete o3 upgrade and getting nearer with it on osd as well.

#7 - 2019-10-14 08:26 - okurz

- Status changed from New to Feedback

- Assignee set to okurz

https://gitlab.suse.de/openqa/salt-states-openqa/merge_requests/198

#8 - 2019-10-23 20:17 - okurz

- Status changed from Feedback to Resolved

so we have a gitlab CI pipeline with rollback possibilities for OSD … I guess this is as good as it gets for now.

#9 - 2020-10-12 13:37 - szarate

- Tracker changed from action to coordination

#10 - 2020-10-12 13:46 - szarate

See for the reason of tracker change: http://mailman.suse.de/mailman/private/qa-sle/2020-October/002722.html