

OsmoBTS - Support #3730

TTCN3 BTS real HW: create jenkins job to build and push ttcn3-bts-test docker image

12/13/2018 12:23 PM - pespin

Status:	Resolved	Start date:	12/13/2018
Priority:	Normal	Due date:	
Assignee:	pespin	% Done:	100%
Category:			
Target version:			
Spec Reference:			

Description

Right now, jenkins job https://jenkins.osmocom.org/jenkins/job/osmo-gsm-tester_gerrit/ calls osmo-gsm-tester.git/ttcn3/jenkins-run.sh which runs the ttcn3-bts-test container:

```
prepare_docker() {
    OLDPWD=$PWD

    # update docker-playground and update the BSC and bsc-test containers (if needed)
    DIR=~/.jenkins/docker-playground
    if [ ! -d "$DIR" ]; then
        mkdir -p ~/.jenkins/ && cd ~/.jenkins
        git clone git://git.osmocom.org/docker-playground
    fi
    cd $DIR
    git remote prune origin; git fetch; git checkout -f -B master origin/master
    cd $DIR/debian-stretch-titan && make
    docker pull laforge/debian-stretch-titan:latest # HACK
    cd $DIR/ttcn3-bts-test && make
    # execute the script to start containers, read results, ...
    #cd $DIR/ttcn3-bts-test && sh -x ./jenkins.sh
    PWD=$OLDPWD
}
```

However, so far I have been building and pushing the image to the docker repository manually. Since we usually introduce new tests and fixes, we need to have that done manually by some jenkins job.

These are the steps required to build it:

```
docker login -u "$MYUSER" registry.sismocom.de # only once required

cd docker-playground.git/ttcn3-bts-test
docker build -t registry.sismocom.de/ttcn3-bts-test .
docker push registry.sismocom.de/ttcn3-bts-test
```

I have the credentials (user+password) to build it which can be applied to a jenkins node, but I'm unsure how to proceed to do this set up.

Which node should I use? How do I apply the credentials? I guess we don't want to run the "docker login" adding the credentials in some public place, rather do it once manually in the designated jenkins node. Are we pushing similar TTCN3 docker containers somewhere? how do we build and push them?

History

#1 - 12/13/2018 03:51 PM - pespin

```
[15:30:33] <pespin> You may know about how to do what I need here: https://osmocom.org/issues/3730
[16:10:04] <osmith@im.sismocom.de> I'll take a look at it
[16:15:56] <osmith@im.sismocom.de> I guess it's easier to chat about this than writing a wall of text in the issue.
[16:16:37] <osmith@im.sismocom.de> I know of two ways to build the docker containers, which we have implemented
```

[16:17:05] <osmith@im.sismocom.de> one is described here: https://osmocom.org/projects/osmocom-servers/wiki/Jenkins_build_verification_jobs - basically osmo-ci.git has a Dockerfile, and the container gets rebuilt whenever osmo-ci.git changes

[16:17:19] <osmith@im.sismocom.de> and the other one is for the ttcn3 tests. all of them get built on demand, without using a shared repo

[16:17:47] <osmith@im.sismocom.de> there's some caching magic in place, which makes Docker skip all steps where nothing has changed

[16:18:01] <osmith@im.sismocom.de> I'll explain with an example

[16:18:51] <osmith@im.sismocom.de> <https://git.osmocom.org/docker-playground/tree/ttcn3-mgw-test/jenkins.sh>

[16:18:57] <osmith@im.sismocom.de> at the top, there is docker_images_require

[16:19:20] <osmith@im.sismocom.de> which is defined in jenkins-common.sh: <https://git.osmocom.org/docker-playground/tree/jenkins-common.sh#n5>

[16:19:48] <osmith@im.sismocom.de> basically it is always running "make" to build the docker container

[16:20:38] <osmith@im.sismocom.de> so in case of ttcn3-mgw-test, it depends among others on osmo-mgw-\$IMAGE_SUFFIX

[16:21:38] <osmith@im.sismocom.de> so if that variable is master, it builds osmo-mgw-master, which is defined here: <https://git.osmocom.org/docker-playground/tree/osmo-mgw-master/Dockerfile>

[16:22:30] <osmith@im.sismocom.de> there are a lot of ADD statements, which are basically the rules when the "RUN" command above can be cached and when they need to be executed again

[16:23:33] <osmith@im.sismocom.de> so when the git repository did not get any new commits, it does not rebuild the software inside the container

[16:23:43] <osmith@im.sismocom.de> *the osmo-mgw.git repository

[16:24:24] <osmith@im.sismocom.de> ...and that's why we can run it every time, it goes through in a second, because everything is cached. and when it's not cached, it builds the container on demand. the container remains on the jenkins slave.

[16:24:59] <osmith@im.sismocom.de> that worked out well, now instead of writing a wall of text into the issue, I wrote it into this chat window. there you go :)

[16:29:25] <pespin> ok thanks

[16:29:30] <pespin> but in this case this cannot be used

[16:29:40] <osmith@im.sismocom.de> none of both methods?

[16:29:57] <pespin> because osmo-gsm-tester_run-prod job is run in OsmoGsmTester MainUnit

[16:30:11] <pespin> and we don't want to build docker containers (specially TTCN3 one) in there

[16:30:19] <osmith@im.sismocom.de> right, that makes sense

[16:30:28] <osmith@im.sismocom.de> but the other method could work

[16:30:29] <pespin> so we need to split building (and pushing) and using (and pulling)

[16:30:35] <pespin> the first one?

[16:30:41] <osmith@im.sismocom.de> yeah, maybe

[16:30:44] <osmith@im.sismocom.de> at least worth looking into

[16:31:03] <pespin> ok I'll have a look

[16:31:54] <osmith@im.sismocom.de> okay, it's also not suitable in that form for your case

[16:32:14] <osmith@im.sismocom.de> as I understand, it builds the image once on all build slaves, it does not push them to a server

[16:32:51] <osmith@im.sismocom.de> so for your case, you'd need a new job in master-builds.yml, with a jenkins.sh that builds and publishes the docker image

[16:33:28] <pespin> yeah, but not sure how to do the credentials stuff

[16:34:08] <osmith@im.sismocom.de> right. you could use the sismocom jenkins server, but I'm not sure if it can trigger on osmocom git repo changes

[16:35:54] <osmith@im.sismocom.de> with some basic research, the jenkins credentials thing seems to be able to store the credentials safely

[16:36:12] <osmith@im.sismocom.de> <https://blog.nimbleci.com/2016/08/31/how-to-build-docker-images-automatically-with-jenkins-pipeline/#comment-3421276058>

#2 - 05/08/2019 06:05 PM - laforge

- Assignee changed from laforge to pespin

We already appear to have a jenkins project "ttcn3-bts-test_docker-registry" at the sismocom jenkins which was setup by @pespin. Does this mean, the ticket is resolved?

#3 - 05/08/2019 06:23 PM - pespin

- Status changed from New to Resolved

- % Done changed from 0 to 100