Automating a functional tests suite in ODL

Sodium DDF April 2019 Guillaume Lambert / Cédric Ollivier
(TransportPCE) feedback
What works well (JUnit)

JUnit Tests are well integrated in Releng
- Console Logs can be watched before the job ends.
- jdk and maven are kept up-to-date
- etc...

But:
- UT cannot be disabled from Releng (does it build w/o UT?)
- macros are available only in master and stable/* branches
- devices simulated must use the mockito framework
- cannot check the whole controller behavior and identify functional regressions
  => « black box approach » or « functional tests »
One solution (among others) to write functional tests: python nose + tox

python proposes « unitests » inspired from Junit

- nose is a python framework. It can be used as a launcher for tests (and much more).
- nose can be used to test the controller from the REST API and against any simulator or equipment.

https://tox.readthedocs.io/en/latest/
Tox is an easy way to automate and standardize tests in python:
- provides python virtualenv
- allow to set up profiles
- allow any command line tool, not just python -> useful to launch sims
- already used by ODL docs / sphynx
Releng has a “gerrit-tox-verify” template with many options
  - gerrit-tox-verify:
    build-timeout: 120
    build-node: centos7-builder-8c-8g

Adaptation needed:

- the build-node does not come with a maven set-up
  -> need to install it from tox
  -> because of virtualenv, don’t follow maven official doc and use symlinks rather than env vars
- functional tests can be quite long (~1h30) => need to increase time-outs
- sims consume memory (by default only 4G RAM) => need to change the build-node image

https://git.opendaylight.org/gerrit/#/c/79983/
https://git.opendaylight.org/gerrit/#/c/78544/
current limitations

Tox profiles cannot be launched separately.

Tox logs are in a separate folder and cannot be watched before the jenkins job ends. We have to way 90min...

- We needed around 4 cores and 6GB RAM but no such image => [centos7-builder-8c-8g.cfg](https://github.com/opendaylight/releng-builder/tree/master/jenkins-config/clouds/openstack/odlvesx)
- The 4GB RAM image sometimes worked but there was only 1GB swap on it…
  => This is prone to memory (re)allocation deadlocks
- No maven installation by default

Criteria acceptance is binary: success or failure. But no all functional tests are expected to pass. Force to success -> still need for a manual check.
Approach inspired from OPNFV functest
General idea behind functest / Xtesting

The main purposes:
- customizing the test environment (Docker minions, Jenkins, etc...)
- testing the controller against real equipment (or a different sim)

How:
Offloading functional tests outside the integration framework with:
- a bot account with -1/0/+1 verified privilege
- a gerrit stream-events connection
Xtesting in a nutshell

- allow the developer to work only on the test suites without diving into CI/CD integration
- check multiple components (see ONAP Orange OpenLab) in the same CI/CD toolchain thanks to a good design
- simplify third-party test integration in a complete OPNFV-based (e.g. de facto standard) CI/CD toolchain
- A key helper from first tests to full end2end service testing
typical deployment scheme

Jenkins

User → Docker Repository → PYTHON → DBMS

Analysis

Job 1
Job 2
...

lab

CLI commands & Management of the power outlet / optical signal

lab → Data files

ROADM A → ROADM B → ROADM C
Thank you