September 25 - 27, 2018
Amsterdam, The Netherlands

OPEN NETWORKING //
Integrate, Automate, Accelerate
LFN Xcommunities Testing

Morgan Richomme & Cedric Ollivier, Orange
Open Networking ecosystem is complex...
Very complex...
(Automatic) Testing is key to

• Validate modules of any size
• Validate End to End solutions
• Prove interoperability between systems
• Ensure sustainability
• Give performance figures and elaborate KPIs
• Ensure backward compatibilities
• Provide trustability evidence
Richness, diversity...

- Lots of testing organizations/committees/cultures
- Lots of frameworks (Robot, python, bash, …)
- Lots of options to integrate in CI (jenkins, travis, gitlab-ci…)
- …. 

or chaos: as many CI/CD chains as subprojects?
Keep it simple
Do not reinvent the wheel
Mutualize
Keep it simple: xtesting
Facilitate the tester life

• ease the development of third-party test cases by offering multiple drivers: Python, Bash, unittest, robot framework and VNF
• manage requirements and offer lightweight Docker images
• handle all interactions with OPNFV CI/CD components (entry points, results, publication, status codes, etc.)

Let the developer only work on the test suites without diving into CI/CD integration
Smoothly integrate all tests together

- simplify test integration in a complete LFN-based CI/CD toolchain (e.g. Jenkins, Testing Containers, Test API and dashboard)
- allow a proper design and verify multiple components in the same CI/CD toolchain (OpenStack, Kubernetes, ONAP, etc.)

Keep Jenkins jobs and certification program simple as well!
### A user story ONAP

<table>
<thead>
<tr>
<th>TIER</th>
<th>CI LOOP</th>
<th>DESCRIPTION</th>
<th>TESTCASE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Set of basic functional tests to validate the ONAP installation.</td>
<td>robot_healthcheck robot_multicloud</td>
</tr>
</tbody>
</table>

Robot Healthcheck: Testing acom components are available via calls.

<table>
<thead>
<tr>
<th>Basic SONIC Health Check</th>
<th>PASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic AAI Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic Policy Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic MSD Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic ASCD Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic APPC Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic Portal Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic Message Router Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic VID Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic Microservices Bus Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>Basic CLAMP Health Check</td>
<td>PASS</td>
</tr>
<tr>
<td>catalog API Health Check</td>
<td>PASS</td>
</tr>
</tbody>
</table>

---

**Testing**

2018-02-13 15:22:38,318 - functest.ci.run_tests - INFO - TESTS TO BE EXECUTED:


2018-02-13 15:22:38,328 - functest.ci.run_tests - INFO - Running test case 'robot_healthcheck'


**Testcase results:**

```

<table>
<thead>
<tr>
<th>status:</th>
<th>PASS</th>
</tr>
</thead>
</table>

2018-02-13 15:22:41,123 - functest.ci.run_tests - INFO - FUNCTEST REPORT:

<table>
<thead>
<tr>
<th>TEST CASE</th>
<th>PROJECT</th>
<th>TIER</th>
<th>DURATION</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>robot_healthcheck robot_dcae robot_multicloud robot_3rdparty</td>
<td>functest onap</td>
<td>00:01</td>
<td>PASS</td>
<td></td>
</tr>
<tr>
<td>functest onap</td>
<td>functest onap</td>
<td>00:00</td>
<td>FAIL</td>
<td></td>
</tr>
<tr>
<td>functest onap</td>
<td>functest onap</td>
<td>00:00</td>
<td>FAIL</td>
<td></td>
</tr>
</tbody>
</table>

```
try it!

run xtesting container:
$ sudo docker run opnfv/xtesting

run xtesting via package (python3):
$ virtualenv xtesting-py3 -p python3
$ . xtesting-py3/bin/activate
$ pip install xtesting
$ sudo xtesting-py3/bin/run_tests -t all
$ deactivate
Coming soon

- deploy the full CI/CD toolchain via docker-compose (Jenkins, Testing Containers, Test API and dashboard)
- offer GNU/Linux packages

All contributions coming from LFN projects are more than welcome!
Do not reinvent the wheel
End to End tests (conformance)

tempest, rally, Shaker, Patrole

testsuite, vnfsdk

Functest  Yardstick

csit  fds

tungstenfabric

OPEN DAYLIGHT  csit  fds

openstack™  End to End tests (conformance)
kubernetes
Upstream first
Do not reinvent the wheel

• Verify OpenStack and Kubernetes deployments via all OPNFV Functest suites (healthcheck, smoke, benchmarking, components, vnf)
• Complete the verification by homemade xtesting containers integrating healthcheck suites and vnf deployments
• Reuse most of OPNFV CI/CD toolchain components (Testing Containers, Test API and dashboard)
Pushing the limits

• Continuously verify OpenStack and Kubernetes deployment
• Run all testcases in parallel once basic tests pass
• Check the possible remaining resources after runs

The current OPNFV gates are fine for checking installers but not for verifying a production environment
Mutualize
Orange Openlab
Mutualize

- Mix & Match LFN project CI/CDs into 1 consistent CI/CD chain
- 1 resource : N LFN deployments
Conclusions
Conclusions

We believe xtesting is valuable for the communities:

• a simple, light and unifying framework without changing community cultures/good or bad habits/design patterns
• the test developers focus only on their tests
• a first step for useful post processing and machine learning as CI/CDs of the different communities generate lots of results

Upstream first & Keep it simple!
Thank you

Visit our booths during ONS
- OpenLab on LFN booth
- NBI/TMF API for ONAP on Orange booth

Other presentations during ONS
26/9 13:50: Lightning Talk: Artificial Intelligence the Next Digital Wave for Telcos - Jamil Chawki, Orange
26/9 14:30 Accelerate the VNF Integration - Jehanne SAVI, Orange
27/9: 11h15 LFN Xcommunities Testing - Morgan Richomme & Cedric Ollivier, Orange
27/9 14:30: Be Active in Networking Open Source when you are a Service Provider - Eric Debeau & Morgan Richomme, Orange
27/9 16:05: SDO + Open Source: When TMF APIs Meet ONAP - René Robert & Matthieu Geerebaert