ONAP Testing – introduction to ONAP SDK

S.Desbureaux (OOM), M.Richomme (Integration)

Prague, 14th of January 2020
Why a Python SDK?

Because it is very useful...
• to create tests
• to automate tests
• to easily interact & perform quick checks on ONAP

Model: OpenStack Python SDK
Another CLI?

- A Java based CLI already exists and is used by a test framework through system calls written in python...
- Bunch of python files created by integration project (https://git.onap.org/testsuite/python-testing-utils/tree) for use cases dealing with robot (not defined as a python SDK),
- SDK in different languages should not be a problem...
A bit of history: from onap-tests

• Since Casablanca some tests are performed using onap-tests (https://gitlab.com/Orange-OpenSource/lfn/onap/onap-tests)
  - Basic python ~SDK + test tool published under Apache v2 on gitlab.com
  - Used in Gating and CI Daily chains to perform end to end tests
  - Developed mainly by Orange with contributions from Exfo and DT
  - Used by third party to onboard/instantiate their VNFs based on their heat template
• But the code quality was relatively poor
  - Poor unit test coverage
  - Poor functional test coverage
  - Lots of copy/paste
  - Poor object-oriented conception
  - Mix SDK (call the API) and the tests

• Decision was taken to heavily refactor onap-tests during Summer 2019 => python-onapsdk
onap-pythonsdk

- Only SDK (still need an onap-tests aside to perform the tests by consuming onap-pythonsdk)
- Higher expectations in term of quality
  - Integrated documentation
  - 100% pylint
  - 100% test coverage
  - Functional tests (with emulators)
  - Better conception (object oriented) for a better maintainability
  - Security scanning
  - License checking
  - Dependency checks
  - Available as a python package: pip install onap-pythonsdk
• Apache v2 hosted in gitlab.com (could be migrated to LF if needed)

https://gitlab.com/Orange-OpenSource/lfn/onap/python-onapsdk
Status

- Onboarding: 90%
- Instantiation (GR-API): 70%

```python
>>> from onapsdk.vendor import Vendor
>>> vendor = Vendor(name="toto")
>>> vendor.onboard()
>>> from onapsdk.vsp import Vsp
>>> vsp = Vsp(name="morganVSP", vendor=toto, package=open('/tmp/ubuntu16.zip','rb'))
>>> vsp.onboard()
>>> from onapsdk.vf import Vf
>>> vf = Vf(name="morganVF", vsp=vsp)
>>> vf.onboard()
>>> from onapsdk.service import Service
>>> service = Service(name="morganService", resources=[vf])
>>> service.onboard()
>>> service.get_tosca()
>>> from onapsdk.ns import NetworkService
>>> ns = NetworkService(name="morganNetworkService", '/tmp/tosca_files/service-X-template.yml')
>>> ns.instantiate()
```
Join us
Thank you!