VNF Validation using OVP Tools

22 April 2020
Rajendra Mishra
rpmishra@aarnanetworks.com
Agenda

What we will share today

- VNF validation on ONAP
- Some challenges we faced while doing OVP tests
  - During Plug test (UNH Lab)
  - Lenovo Lab (Beyond the plug test)
- Brief information about what was tested
VNF Testing & Validation in ONAP

- Test frameworks in ONAP
  - VVP : VNF Validation Program
  - OVP : OPNFV Verification Program
  - VTP : VNF Test Platform

- Scope of today’s presentation will be
  - VVP
  - OVP
Static Validation aka VVP

- This is one of the first tests for ONAP VNF validation.
- It takes VNF HEAT template files and does static validation on it.
- There are nearly 127 tests that are run to validate the correctness of the HEAT templates.
- The test is written using python and can be run on laptop.
Typical issues found in VVP

- Unused parameters in HEAT template are flagged as error, they need to be eliminated.
- Some parameters in template (e.g. network ports) should be unique. We used `str_replace` to associate `VNF_NAME` or `INSTANCE ID` with the port.
OVP : OPNFV Validation program

- OVP is the superset of VNF testing
- OVP is built using robot script
- The input is a DIR structure that should have
  - VNF HEAT template files
  - SDNC preload files (in json format)
  - Openstack access information like subscriber, tenant, region etc.
System Configuration

- HW details
  - Lenovo ThinkSystem SR650 servers
    - (Dual-Socket Intel Xeon 6152 CPU, 88 cores, 2TB SSD)
  - Lenovo ThinkSystem NE2572/NE0152T switches
- Linux Centos 7.6
- Aarna Networks ONAP Distribution (ANOD)

SERVERS Provided By Lenovo Labs
While running the tests we use the robot script which does the following

- STATIC Validation using vvp tests
- Onboard the VNF to ONAP
- Instantiate it in Openstack

```
rp@rp-VirtualBox:~/aarna/om5/kubernetes/robot$ ./instantiate-k8s.sh -h
./instantiate-k8s.sh [options]

required:
-n, --namespace <namespace>  namespace that robot pod is running under.
-f, --folder <folder>        path to folder containing heat templates, preloads, and vnf-details.json.

additional options:
-p, --poll

This script executes the VNF instantiation robot testsuite.
- It copies the VNF folder to the robot container that is part of the ONAP deployment.
- It models, distributes, and instantiates a heat-based VNF.
- It copies the logs to an output directory, and creates a tarball for upload to the OVP portal.
```

rp@rp-VirtualBox:~/aarna/om5/kubernetes/robot$
Once the onboarding is complete, verification checks if the stack was created successfully.

It validates all the parameters passed in template environment with what is returned by Openstack.

A report is generated. Report is a json file that contains the checksum of VNF templates along with the tests that passed successfully.
Accomplishments

- We got the ONAP VVP testing (OOM Robot) running on two platforms.
- Worked on testing 3 commercial VNFs through these systems.
- Onboarded one of the VNFs through ONAP Dublin release.
- VNF static (template) validation passed on all 3 VNFs.

Challenges

- OPNFV XCI OpenStack setup provides HTTPS for OpenStack API by default, using self-signed certificates. Within ONAP, this requires adding the self-signed CA to multiple pods.
- During ONAP deployment, the authentication keys should have been stored within correct formats for SO / Robot / etc. However, this seems to have failed during the install and required manual correction.
- Repeatedly running e.g. the robot scripts while debugging can leak state into ONAP that requires manually cleaning databases. The option to rollback changes or having a “wipe clean” script for A&AI would be very useful.
- Initialization of values for ONAP (i.e. subscriber, cloudowner, line of business, etc.) isn't clearly defined in the process, and if / who is responsible for setting those values. For example “demo-k8s.sh onap init” will setup / provide one set of values, while the “instantiate-k8s.sh” for the VVP testing may require similar ones again.
- Robot VVP script failures had to wait for timeout (i.e. script stopped) before logs became available to debug the issue.
- Need to get some support from community to provide TOSCA based VNFs to run through the testing process.
We faced few issues while running the tests for a commercial VNF

- We started the tests with Dublin version but had to switch to El Alto due to the dependencies for later versions of Robot and other modules
- OVP runs on El Alto and later version. Getting a stable El Alto in a single server was a bit of challenge
  - We faced issue regarding Openstack authentication (CERTIFICATE)
  - Some pods do not come up (like SDC-BE) after install
  - The timeout values at helm install time has to be tuned in for different setups
- We need to make sure that the templates pass the VVP tests before running OVP
- All resources used by templates like networks, images, flavors etc. should be present in Openstack before the test is run
- Some specific commercial VNF might be using HPA features (e.g. SRIOV) so the tests has to be run on Openstack that supports all the features used.
- We ran into two bugs in OVP framework. The same should be fixed in newer release of ONAP.
Details of bugs

- Admin tenant value need to be updated manually.
  
  Workaround in below mail.
  
  https://www.mail-archive.com/onap-discuss@lists.onap.org/msg18464.html

  Jira Ticket: https://jira.onap.org/browse/TEST-232

- VNF Checksum calculation code in OVP runs into infinite loop.
  
  More details with workaround in below mail.
  
  https://www.mail-archive.com/onap-discuss@lists.onap.org/msg18630.html

  Jira Ticket: https://jira.onap.org/browse/TEST-233
Useful Links


https://wiki.lfnetworking.org/display/LN/LFN+Developer+and+Testing+Forum+Jan+2020+OVP+VNF+Hacking+Track

https://docs.google.com/document/d/1Rukye8ARDnficNp85fIt0kkKKFK1n0uuj63JXXC--3Q/edit#heading=h.qrjhy43evokx

https://wiki.onap.org/display/DW/OVP-VTP

https://wiki.onap.org/pages/viewpage.action?pageId=68546123

https://vnf-verified.lfnetworking.org/

https://docs.onap.org/en/elalto/submodules/oom.git/docs/oom_quickstart_guide.html
Thank You !