

Jan 2020 LFN Developer & Testing Event—OVP VNF Hacking Track Test Plan

Version	1.0
Authors	Amar Kapadia, Lincoln Lavoie, Pierre Lynch, Rabi Abdel, Sriram Rupanagunta

Overview

The OPNFV Verification Program ([OVP](#)) is an open source, Linux Foundation Networking community-led compliance and verification program to demonstrate the readiness and availability of commercial NFV products and services, including NFVI and VNFs, using OPNFV and ONAP components.

At the [January 2020 LFN Developer & Testing Forum](#), community members plan to support a “OVP VNF Hacking Track” where VNF vendors can conduct interoperability testing with ONAP and OPNFV. The participants in this track will be software engineers from VNF vendor companies. Participants will have access to onsite ONAP, CNTT, and OPNFV experts.

Prerequisites

You will need to:

- Have a deep understanding and access to your VNF software and optional VNFM if needed
- Have working knowledge of OpenStack
- Be comfortable with OpenStack HEAT or TOSCA modeling languages
- Have elementary knowledge of ONAP and the VNF Requirements subproject

Tracks

There are two mutually exclusive VNF interoperability testing tracks: direct or indirect.

Direct

In this mode, ONAP directly interacts with the VNF without any intermediary software. This mode requires a VNF package (descriptor) to be written in the OpenStack Heat modeling language.

Indirect

In this mode, ONAP interacts with the VNF via an external sVNFM/EMS. This mode requires the VNF package (descriptor) to be available in the TOSCA modeling language.

Pretesting

In this stage, you should:

1. Choose one of the above tracks.
2. Run your VNF package through the current OVP [compliance](#) testing. This testing is static-only and requires neither a running instance of ONAP or OPNFV (OpenStack). Of course, you can come to the Plugfest without this step done, but we have found that attendees end up wasting a lot of time if this is not done before hand.

Instructions on how to run static compliance tests for Heat or TOSCA are available [here](#) (see Appendix A for more).

3. Install your sVNFM (if required) in an environment that will be reachable by the Plugfest ONAP instance(s).
 - a. Details on whether the sVNFM needs to be reachable by a public IP or via VPN are still being worked on (**TO DO**)
4. Prepare your image so that it is ready for upload to the Plugfest OpenStack environment.

Onsite Resources

ONAP+OPNFV instance availability:

- 1x UNH-IOL hosted ONAP EI Alto instance
- Second instance location being determined (**TO DO**)

External connectivity to reach sVNFMs as needed

Procedure to upload Glance images:

Flavors available:

Baseline Testing

The baseline testing will run pre-release basic OVP validation testing on the VNF. This basic validation includes:

- Onboarding of the VNF onto ONAP
- Creation of a simple network service with just the single VNF
- Deploying the network service (VNF) onto OPNFV; initial orchestration and deployment performed via ONAP

The current test requirements for OVP VNF Validation testing are defined within the ONAP VNF-RQTS project and have been updated as part of the ONAP EI Alto release. You can view those requirements [here](#).

This testing will be done by running scripts (CLI or REST API).

Advanced Testing (OPTIONAL)

If you complete the basic testing step above, you can optionally do additional advanced testing. Some of this advanced testing will have automation available, while the rest will be manual. If you choose any of these, we will pair you up with an ONAP expert. In order of difficulty:

Configuration testing

The VNF will be tested for day 1/2 configuration. This will be performed via the ONAP CDT tool.

Create a CDT template for config-modify operation of one parameter in the VNF
Onboard and deploy VNF/NS in ONAP (if not already done)
NETCONF-mount the VNF that needs be configured
Update the parameter using CDT
Run APP-C API to make the changes to the parameter have indeed been made
Read the parameter value to check for the updates

Scale-out testing

In this test, we will perform manually triggered scale-out of the VNF.

Create a CDT template for config-scaleout of the VNF
Onboard and deploy VNF/NS in ONAP (if not already done)
NETCONF-mount the VNF that needs be configured
Run APP-C API to trigger scale-out operation
Check if the VNF has been scaled out, by performing health-check operation of the newly instantiated VNF
Check if the VNF has been instantiated using OpenStack Nova command

HPA testing

This testing is only applicable for Heat based VNFs. Along with the VNF, there will be a Policy file describing the HPA attributes required by the VNF. ONAP orchestration will match the attributes with those of the destination cloud.

Onboard VNF/NS in ONAP (if not already done)
Create HPA Policy with requirements on vcpu count of 'n'
Create two clouds, with one of them (cloud-1) having compute resources of less than 'n', and another (cloud-2) with more resources
Deploy NS using this HPA Policy
Check to make sure the VNF is deployed on cloud-2

Change management testing

In this test, we will upgrade a VNF.

Onboard VNF/NS in ONAP (if not already done)
Upgrade VNF version
Check to make sure the version is updated?

Monitoring/CLAMP testing

This testing requires a VES agent in the VNF/VNFM. If the VES agent is available and configured correctly, then monitoring and closed-loop automation can be tested. We will create a simple closed loop that will reset the VNF.

Onboard VNF/NS in ONAP (if not already done)
Create Policy to respond to one variable in VNF (var1) reaching a threshold (th1)
Start VNF operation that results in var1 reaching th1
Check the events in DMaaP to make sure event (ev1) is generated
Create a closed control loop that changes parameter (param1) on receiving event (ev1)
Check VNF (using any means) if param1 is updated

Conclusion

As a VNF vendor, you can do as little or as much OVP VNF interoperability testing as you would like in a safe judgement free zone with access to open source ONAP and OPNFV software and experts on those software projects.

Appendix A

The documentation for the OVP VNF Compliance testing is available here: TBD

VVP Testing

For Heat based VNF compliance testing, here are the steps:

```
git clone https://github.com/onap/vvp-validation-scripts
cd vvp-validation-scripts
virtualenv -p python3 vvp_env
source vvp_env/bin/activate
pip install -r requirements.txt
cd ice_validator/tests
pytest --tap-stream --template-directory=<Directory where the template
files are present>
```

Results will be in ice_validator/output/ directory