

# VNF Validation Minimum Viable Product

This is a work in progress to define the minimum viable/**valuable** product (MVP) for a program to validate the life-cycle of a VNF running "in/with" ONAP.

The MVP definition should not discourage anyone from contributing to other projects or efforts related to VNF testing, but should help guide developer priorities for test frameworks.

## Requirements & Deliverables

1. The goal of the MVP is to specify a suite of VNF validation tests and associated test infrastructure that can be developed within the time-frame of the E-release of ONAP.
2. Tests will focus on **on-boarding and instantiation** of the VNF in/within ONAP.
3. Tests will focus on HEAT and TOSCA based VNFs.
4. Test will leverage the SDNC and APPC, VFC controllers.
5. Tests will use the VIM that is provided with ONAP i.e. a generic version of OpenStack. (a future program update / release may migrate to other VIMs).
6. VNFs are validated against the specific release of ONAP (i.e. VNF is validated against ONAP E-release or F-release).
7. VNFs must also pass the current VNF criteria defined by ONAP for the initial release of VNF compliance testing.
8. Testing uses existing interfaces into ONAP for stimulus / response to "drive" the tests (i.e. avoid creating new requirements / interfaces for the ONAP at large project).

## What is needed / Work to do

1. ONAP release must be readily deploy-able, to allow a test framework to run on the deployment and test a VNF.
2. ONAP deployment must be reproducible to ensure VNF tests are conducted in a uniform environment.
3. Definition of life cycle requirements (i.e. what the test cases validate)
  - a. Both HEAT & TOSCA
    - i. What does a mean (formally for a test case requirement) to "instantiate" (or startup) a VNF?



from Victor, Definition of VNF onboarding for TOSCA based VNF.

- Verify the VNF package by using VNFSdk compliance check test cases.
- upload a VNF package to ONAP system
- verify the VNF Package already exist on the ONAP system.



from Victor, Definition of VNF instantiation for TOSCA based VNF.

- Trigger an "instantiate VNF" operation to Controller
- verify that the requested grant for the "instantiate VNF" operation has been approved by the controller
- VNF related software images have been successfully added to the image repository managed by the VIM or already exist on the VIM system (Currently, ONAP is manually doing this.)
- Verify that the requested virtualised resources have been allocated by the VIM according to the VNFD
- Verify that virtualised resource allocation constraints have been met by querying the VIM
- Verify that any existing virtualised resources have not been affected by the allocation of the new virtualised resources by querying the VIM
- Verify that the VNF instance resources are visible on the controller

- ii. Does instantiation include configuration?
  1. Victor: both heat and Tosca template could include the inject-file/user-data(cloud-init) as the day0 configuration file of a VNF.
  2. Trevor: I would argue against configuration for the MVP phase. There are a variety of options to execution post-instantiation configuration management (Ansible, Chef, etc.). Since there are multiple options for a compliant VNF to utilize it would not make sense to offer only a subset of options, and offering any option would involve setup, configuration, and execution of the configuration mgmt option itself.
- iii. Does instantiation include health check?
  1. Victor: Could be yes, we could use the query interface (query the VNF detail info) to implement the health-check func.
- iv. HEAT
  1. [Trevor Lovett](#) & [Ryan Hallahan](#) will help with this.
  2. The following is a DRAFT - needs further review and input
  3. **Prerequisite Decision Points**
    - a. Are we planning for decentralized (VNF Provider must setup their own ONAP instance and perform their own testing) or centralized testing (VNF Provider is submitting their VNF and associated settings for testing) model? Assuming the former, but need confirmation as the latter has additional automation and other concerns to care for (network creation, image registration, etc.)
  4. **Assumptions**
    - a. Test scripts will utilize a default license model in SDC
    - b. Compliance with test suite assumes the VNF has already has compliant packaging certified

- c. If utilizing full automated testing, then assume the Service model will only include a single VF for instantiation. It would likely be too difficult to automatically create a multi-VF service in the first phase, and not necessary for the scope of testing
- d. Test Input will include the preload information in some TBD file format that can be loaded into SDNC.
5. **Implementation Options**
  - a. Within ONAP today, the Integration Testsuite generally has all the building blocks to automate and execute the steps below. It would need some modification to handle a more arbitrary VNF as it is not used in this way today, but based on preliminary analysis this looks achievable.
  - b. Orange Python Framework - This is another option that can be considered. It looks to handle the instantiation use case, and potentially others.
6. **Potential Test Flow**
  - a. Initialize Vendor and Category Information
  - b. Create the VSP in SDC
  - c. Upload Heat Archive
  - d. (Optional) Assign any Unassigned Files to Artifacts
  - e. Validate the VSP and ensure now Errors exist (warnings are OK)
  - f. Assign the Vendor License Model to the VSP (assumes a single VLM for testing purposes)
  - g. Create the Virtual Function
    - i. Import the VSP (find using Name or ID from prior steps)
    - ii. Set name of VF (auto-assign or make input into test script), contact and other required fields
  - h. Create Service
    - i. Set Name (auto-assign based on VSP or make input into test script)
    - ii. Assign required or optional fields based on test script input
    - iii. Assign VF to the Service Model
  - i. Distribute the Service Model and validate successful Distribution
  - j. Submit Preloads to SDNC
  - k. Trigger Instantiation of Base Module from VID (NOTE: Need to see how we handle multi-module VNFs - presumably we can query this information and instantiate each individually)
  - l. Verify successful instantiation
  - m. Health-check TBD - needs further discussion
- v. **TOSCA**
  1. [Weitao Gao](#) will help with this.



from Victor, Definition of VNF instantiation for TOSCA based VNF.

- Trigger an "instantiate VNF" operation to Controller
- verify that the requested grant for the "instantiate VNF" operation has been approved by the controller
- VNF related software images have been successfully added to the image repository managed by the VIM or already exist on the VIM system(Currently, ONAP is manually doing this.)
- Verify that the requested virtualised resources have been allocated by the VIM according to the VNFD
- Verify that virtualised resource allocation constraints have been met by querying the VIM
- Verify that any existing virtualised resources have not been affected by the allocation of the new virtualised resources by querying the VIM
- Verify that the VNF instance resources are visible on the controller

- b. Test case description template for specifying VNF validation test purpose, implementation steps and pass/fail criteria.
- c. Definition of the set of ONAP components and their configuration required for the testing (ONAP profile used for testing).
- d. Definition of test infrastructure requirements needed for testing (i.e. hardware with compute / network / storage and pod requirements to run testing).
- e. Test tooling that drives testing through existing ONAP interfaces.
  - i. HEAT
  - ii. TOSCA

## What is needed / Work to do Matrix

### Test Life Cycle

| # | Step (Common)                                                                          | HEAT Specific | TOSCA Specific                                       |
|---|----------------------------------------------------------------------------------------|---------------|------------------------------------------------------|
| 1 | Initialize Vendor and Category Information                                             | via SDC       | N/A- already in SOL001 VNF Descriptor                |
| 2 | Create the VSP in SDC                                                                  | via SDC       | via SDC                                              |
| 3 | Upload Archive                                                                         | Heat Archive  | ETSI SOL004 CSAR File                                |
| 4 | (Optional) Assign any Unassigned Files to Artifacts                                    | via SDC       | Error or warning since package should match manifest |
| 5 | Validate the VSP and ensure no Errors exist (warnings are OK)                          | via SDC       | via VNFSDK                                           |
| 6 | Assign the Vendor License Model to the VSP (assumes a single VLM for testing purposes) | via SDC       | via SDC                                              |
| 7 | Create Virtual Function - Import the VSP (find using Name or ID from prior steps)      | via SDC       | N/A- already in SOL001 VNF Descriptor                |

|    |                                                                                                                                                                                |                                                                      |                                                                  |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|------------------------------------------------------------------|
| 8  | Create Virtual Function - Set name of VF (auto-assign or make input into test script), contact and other required fields                                                       | via SDC                                                              | N/A- already in SOL001 VNF Descriptor                            |
| 9  | Create Service - Set Name (auto-assign based on VSP or make input into test script)                                                                                            | via SDC                                                              | via SDC                                                          |
| 10 | Create Service - Assign required or optional fields based on test script input                                                                                                 | via SDC                                                              | via SDC                                                          |
| 11 | Create Service - Assign VF/VNF to the Service Model                                                                                                                            | via SDC                                                              | via SDC                                                          |
| 12 | Distribute the Service Model and validate successful Distribution                                                                                                              | via SDC DMaaP                                                        | via SDC DMaaP                                                    |
| 13 | Submit Preloads                                                                                                                                                                | via SDNC                                                             | via SDNC                                                         |
| 14 | Trigger Instantiation of Base Module from VID (NOTE: Need to see how we handle multi-module VNFs - presumably we can query this information and instantiate each individually) | via VID                                                              | via VID to SO & SOL003 adapter or VFC & SOL003 adapter           |
| 15 | Verify successful instantiation                                                                                                                                                | Verify Heat Stack Create Successful<br><br>Ping Ports on OAM network | Verify VNF created successfully<br><br>Ping ports on OAM network |

## Items to do

Based on a review with the ONAP Integration team the test suite Robot scripts provide the majority of the building blocks to perform the automation required for Heat-based VNFs this effort. There is still work to adapt the existing scripts to handle a generic VNF vs. the predefined demo VNFs currently used. The amount of effort on a per function basis as laid out in the table is not known at this time, but the overall effort does look to be achievable in the El Alto time frame.

| # | item                                    | HEAT specifics                                                       | exist? | resources needed                                  | TOSCA specifics                                                      | exist? | resources needed |
|---|-----------------------------------------|----------------------------------------------------------------------|--------|---------------------------------------------------|----------------------------------------------------------------------|--------|------------------|
| 1 | Update VNFREQTS for LCM definition      | Requirements for VNF "life-cycle" will be the same for HEAT / TOSCA. | 70%    | VNFREQTS Team                                     | Requirements for VNF "life-cycle" will be the same for HEAT / TOSCA. | 50%    | VNFREQTS Team    |
| 2 | Automation Script(s) to on-board VSP    | Integration TestSuite                                                | Yes    | Contributions to Integration project by VVP team. | Victor: Investigating to reuse the existing scripts.                 | ~80%   | VNFSDK Team      |
| 3 | Automation Script(s) to Create VF       | Integration TestSuite                                                | Yes    | Contributions to Integration project by VVP team. | Victor: Investigating to reuse the existing scripts.                 | ~80%   | VNFSDK Team      |
| 4 | Automation Script(s) to Create Service  | Integration TestSuite                                                | Yes    | Contributions to Integration project by VVP team. | Victor: Investigating to reuse the existing scripts.                 | ~80%   | VNFSDK Team      |
| 5 | Automation Script(s) to Submit Preloads | Integration TestSuite                                                | Yes    | Contributions to Integration project by VVP team. | SDN-C Specific Operation: TOSCA could be ignored                     | N/A    |                  |
| 7 | Automation Script(s) to Instantiate VNF | Integration TestSuite                                                | Yes    | Contributions to Integration project by VVP team. | Need to develop new scripts                                          | No     | VNFSDK Team      |
| 8 | Automation Script(s) to Healthcheck VNF | N/A - not planned for this phase                                     | N/A    |                                                   | Nice to have                                                         | N/A    |                  |
| 9 | Clean up after test(s)                  | Implemented directly in TestSuite                                    |        | TestSuite Team                                    | Implemented directly in TestSuite                                    |        | TestSuite Team   |

## Open Questions

- Can the test requirements or definitions (procedures) be pulled from, or reuse, the [ETSI TST-0007](#)
  - What are the integration and testing interfaces that are currently available, i.e. used by the integration team / gating team?

## Definition of Done / Success Measures

- Tests can readily be run, with high level of repeatability.
  - Level of complexity is manageable by end users (i.e. ease of ONAP deployment + test cases).

## EUAG Feedback

Please place your feedback here, as needed.

- Feedback from CMCC: The VNFD we are using in our company are all **TOSCA-based**. Also, we are using VFC for VNF LCM. We suggest to update the 3rd item and the 4th item in "Requirements & Deliverables" to **"Tests will focus on HEAT based VNFs and TOSCA based VNFs"** and **"Test will leverage the SDNC, APPC and VFC controllers"**.
  - Verizon feedback: As mentioned in the comment below we are using **TOSCA** based VNF-D as well. We would like to extend the requirements to include **SOL004/SOL001 VNFs** using the **SDCSOSOL003 Adapter External VNF/VNF path** or the **SDC SO SOL005 Adapter VF-C VNF** path.
  - China Telecom Feedback: We have both HEAT based and TOSCA based VNFDs in the DEMO running on our testbed. It will be great if TOSCA based VNFs could be added in the Requirement, providing the resources are available.

- c. ChinaUnicom Feedback Our company also has the strong demands for TOSCA based VNF and we hope that it could be included into the scope.

## Timeline

| Date                   | Deliverable                                                                                                             |
|------------------------|-------------------------------------------------------------------------------------------------------------------------|
| April 19, 2019         | MVP agreed by the CVC.                                                                                                  |
| April 23, 2019         | Presentation of MVP to LFN EUAG during teleconference                                                                   |
| End April              | MVP agreed / finalized (feature freeze)                                                                                 |
| Late April / Early May | Meetings with development/technical teams to determine what currently exists and what needs to be proposed as new work. |
| Late May               | Development plans finalized with technical teams.                                                                       |
| June 13, 2019          | ONAP E-release M1                                                                                                       |
| June - July            | VNF requirements created for life-cycle                                                                                 |
| July 18, 2019          | ONAP E-release M2/M3                                                                                                    |
| July                   | Test case development, per requirements set                                                                             |
| August                 | Test tooling development                                                                                                |
| August 29, 2019        | ONAP E-release                                                                                                          |
| September              | Beta testing from E-release, requirements frozen / completed, test case and tooling bug fixes only                      |
| October                | Beta conclusions, first VNFs publicly listed as passing the validation testing                                          |

## External Resources

<https://wiki.onap.org/display/DW/OVP+LCM+Support>

[Testing framework comparison](#)