Dovetail Test Case Requirements

This page is intended as a working draft of how we will look to leverage and develop test cases for the dovetail project.

Dovetail Test Suite Purpose and Goals

The dovetail test suite is intended to provide a method for validating the interfaces and behaviors of an NFVi platform according to the expected capabilities exposed in an OPNFV NVFi instance, and to provide a baseline of functionality to enable VNF portability across different OPNFV NFVi instances. All dovetail tests will be available in open source and will be developed on readily available open source test frameworks.

Working with the ETSI NFV TST 001 reference: http://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/001/01.01.01_60/gs_NFV-TST001v010101p.pdf
The Dovetail project will focus on tests validating a scope for a System Under Test (SUT) associated with Chapter 4.9 - NFV Infrastructure + VIM Under Test, as adapted for OPNFV (see figure below). The test suite will also define preconditions and assumptions about the state of any platform under evaluation. The test suite must not require access to OPNFV infrastructre or resources in order to pass.
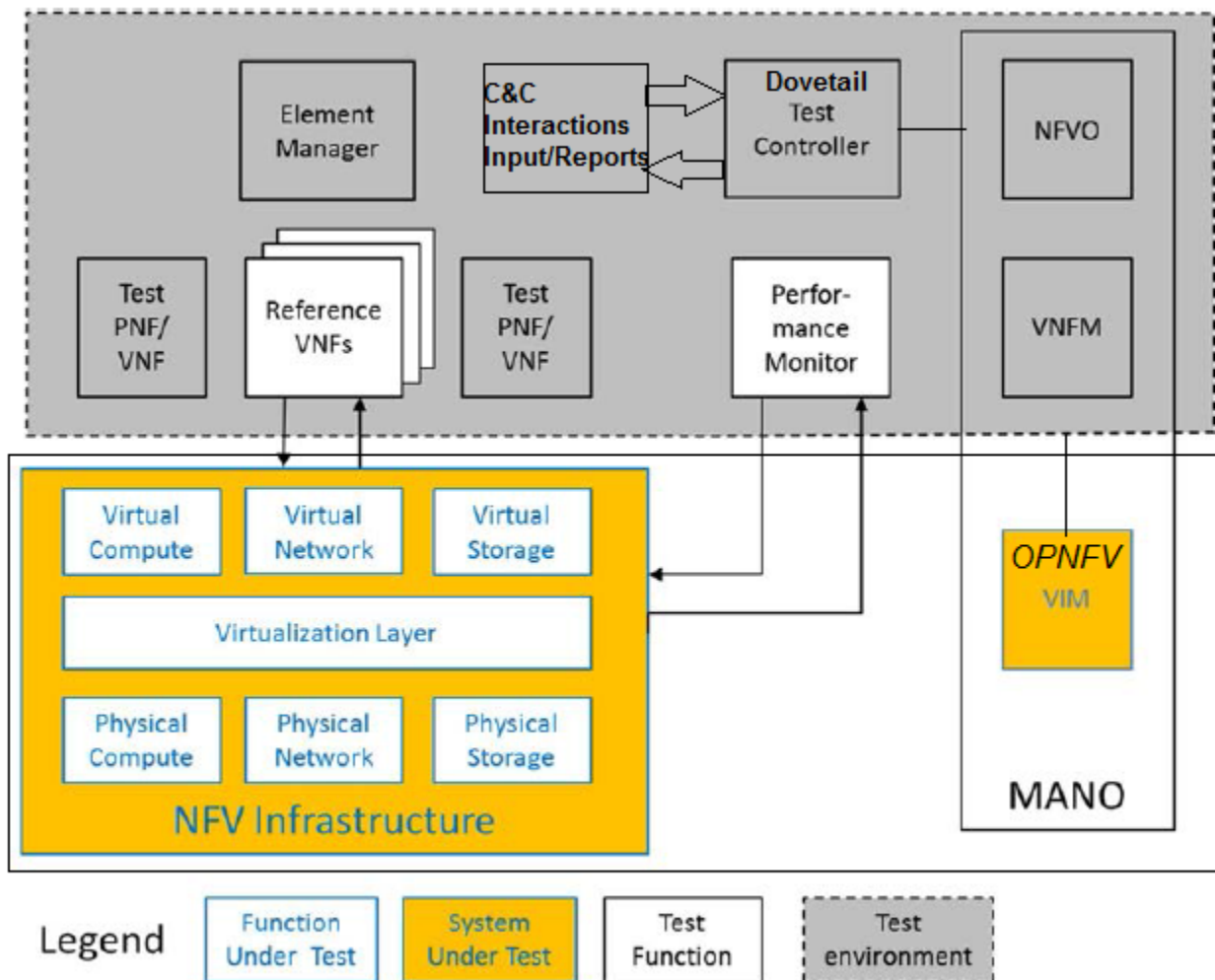

Figure : Functional architecture for (OPNFV) NFVI + VIM Under Test

Test case requirements

The following requirements should be fulfilled for all tests added to the Dovetail test suite:

- Test cases should favour implementation of a published standard interface for validation
  - Where a compliance test suite exists for components of the SUT, this test suite should generally be considered as a baseline for Dovetail testing
  - Where no standard is available provide API support references
  - If a standard exists and is not followed, an exemption is required
- The following things must be documented for the test case:
  - Use case specification
  - Test preconditions
  - Basic test flow execution descriptor
  - Post conditions and pass fail criteria
- The following things may be documented for the test case:
  - Parameter border test cases descriptions
  - Fault/Error test case descriptions
- Test cases must pass on OPNFV reference deployments
  - Tests must not require a specific NFVi platform composition or installation tool
  - Tests must not require unmerged patches to the relevant upstream projects
  - Tests must not require features or code which are out of scope for the latest release of the OPNFV project

New test case proposals should complete a Dovetail test case worksheet to ensure that all of these considerations are met before the test case is approved for inclusion in the Dovetail test suite.

Considerations for the Test Strategy Document

There is a draft test strategy document for OPNFV in preparation (owner: Chris Price).

- JIRA issue: https://jira.opnfv.org/projects/DOVETAIL/issues/DOVETAIL-352
- Gerrit review: https://gerrit.opnfv.org/gerrit/#/c/30811/

Dovetail Test Suite Structure

A dovetail test suite should have the following overall components and structure: (stolen, if simplified a little, from IEEE)

- Test Plan
  - The test plan should describe how the test will proceed, who will do the testing and what will be tested.
- Test Design Specification
  - The design spec outlines what needs to be tested including such information as applicable standards, requirements, and tooling.
- Test Case Specification
  - The Test Case spec describes the purpose of a test, required inputs and expected results, including step-by-step procedures pass/fail criteria.
- Test Procedure Specification
  - Describes how to run the test, preconditions and procedural steps to be followed.
- Test Log
  - A log of the tests run, by whom, when, and the results of the tests.
- Test Report
  - The test report describes the tests that were run, any failures and associated bugs, detailing deviations from expectation.
- Test Summary Report
  - Summary of all results including a test result assessment (to be defined).

Dovetail Test Suite Naming Convention

Test case naming and structure for dovetail, external facing naming sequences for compliance and certification test cases.

Dovetail Test Result Compilation, Storage and Authentication

Test execution identification, results evaluation, storage, identification and security for dovetail compliance and certification test cases.

Phasing the Dovetail Development Effort

While not all tests will be possible to develop at once the following approach is proposed for the development of dovetail test suites in a structured manner.

Dovetail phase 1

Dovetail should initially set out to provide validation of interfaces and behaviors common to an OPNFV NVFi. This can be seen as a set of test cases that evaluate if a NVFi implementation is able to achieve a steady operational state covering the common behaviors expected of an OPNFV NFVi. In this case the dovetail tests will focus on a SUT definition of VNFi & VIM as described in 4.9 of the ETSI NFV TST 001 specification.

Dovetail phase 2

Dovetail should further establish a set of test suites that validate additional desired OPNFV VNFi behaviours. This may include for instance, deployment specific capabilities for edge or remote installations. It may include the validation of functionality that is not yet common to all OPNFV VNFi scenario's.

In phase 2 it may also be possible that dovetail provides such services as application test suites to validate the behavior of applications in preparation for deployment on an OPNFV VNFi. This may result in the definition of new SUT scopes for dovetail as described in of the ETSI NFV TST 001 specification.

Dovetail phase 3

hongbo: the same as those defined in the phase 1 and phase 2

Annotated brainstorming/requirements proposals

Additional Requirements / Assumptions (not yet agreed by Dovetail Group)

These are additional requirements and assumptions that should be considered by the group. This agreement / issue is being tracked in JIRA, under DOVETAIL-352. As these are agreed, they should be moved above into the full list. Once the story is completed, this section can be deleted.

Assumptions

1. Tests start from (use) an already installed / deployed OPNFV platform. OPNFV deployment/install testing is not a target of the program (that is for CI).
   a. *DN: Dovetail should be able to test platforms which are not OPNFV scenarios - we have a requirement that OPNFV scenarios should be able to pass the test suite, which ensures that only features in scope for OPNFV can be included*

Requirements

1. All test cases must be fully documented, in a common format, clearly identifying the test procedure and expected results / metrics to determine a "pass" or "fail" result for the test case.
   a. *DN: We currently list a set of things which must be documented for test cases - is this insufficient, in combination with the test strategy document?*
      i. *lylavoie - No, we need to have the actual list of what things are tested in Dovetail, and how those things are tested. Otherwise, how to can we even begin to know if the tool test the things we think it does (i.e. validate the tool).*
      ii. *DN: I agree, but the scope of the test suite is a topic for the test strategy document, and not part of the decision criteria for a specific, individual test. Test case purpose and pass/fail criteria are part of the documentation requirements.*
2. Tests and tool must support / run on both vanilla OPNFV and commercial OPNFV based solution (i.e. the tests and tool can not use interfaces or hooks that are internal to OPNFV, i.e. something during deployment / install / etc.).
   a. *DN: Again, there is already a requirement thsat tests pass on reference OPNFV deployment scenarios*
      i. *lylavoie: Yes, but it can not do that by requiring access to something "under the hood," this might be obvious, but it's an important requirement for Dovetail developers to know.*
      ii. *DN: Good point. We do have that test cases must use public standard interfaces and APIs.*
3. Tests and tool must run independent of installer (Apex, Joid, Compass) and architecture (Intel / ARM).
   a. *DN: This is already in the requirements: "Tests must not require a specific NFVi platform composition or installation tool"*
4. Tests and tool must run independent of specific OPNFV components, allowing different components to "swap in". An example would be using a different storage than Ceph.
   a. *DN: This is also covered by the above test requirement*
5. Tool / Tests must be validated for purpose, beyond running on the platform (this may require each test to be run with both an expected positive and negative outcome, to validate the test/tool for that case).
   a. *DN: I do not understand what this proposal refers to*
      i. *lylavoie - The tool and program need must be validated. For example, if a test case purpose is to verify a specific API is implemented or functions is a specific way, we need to verify the test tool does actually test that API/function. Put differently, we need to check the test tool doesn't false pass or false fail devices. This is far beyond just a normal CI type test (i.e. did it compile and pass some unit tests).*
      ii. *DN: This is part of the test strategy document rather than part of the requirements for a specific test*
6. Tests should focus on functionality and not performance.
   a. Performance test output could be built in as "for information only," but must not carry pass/fail metrics.
   b. *DN: This is covered in the CVP already*

Additional brainstorming
This section is intended to help us discuss and collaboratively edit a set of guiding principles for the dovetail program in OPNFV.

- If you want to edit this page, please make sure that you add items/bullets/sub-bullets with you name identified as e.g. [bryan] so that we can track who has contributed the perspectives. We should work together to tweak the items/descriptions, so perspectives and responses added by others should be addressed asap, so that this becomes an effective tool for driving consensus.

- If we need a more long-winded/threaded discussion on the topic, please add an item to https://techdiscuss.opnfv.org/.

- This is itself a strawman proposal for how we can discuss this important topic. Alternatives are welcome, although the basic goal is that we can come to collaborative consensus as quickly as possible, with minimal barriers to participation in the discussion.

1) Focus on OPNFV role and unique aspects, e.g.

- Integrated reference platforms

- Component diversity

- OPNFV-driven features

- The basic assumption is that the scope of compliance testing falls within the included features of a given OPNFV release (as its upper bound)
- *DN: Additional clarification added that OPNFV is the upper bound: "Tests must not require features or code which are out of scope for the latest release of the OPNFV project"*

2) Leverage upstream frameworks

- wherever possible, verify components using upstream frameworks as a complete package (vs curating specific tests)

- work proactively to further develop those frameworks to meet OPNFV needs.
- *DN: Requirements modified to clarify: "Where a compliance test suite exists for components of the SUT, this test suite should generally be considered as a baseline for Dovetail testing" - there was a feeling that we could not wholesale accept upstream frameworks without question, but the goal is to ensure that OPNFV compliant solutions are also compliant to component certification requirements*

3) Deliver clear value: release only tests that add clear value to SUT providers, end-users, and the community

- *DN: Not easily actionable for test case requirements*

4) Agile development with master and stable releases

- fast rollout with minimum viable baseline, and continuous agile extension for new tests

- Support SUT assessment against master and stable releases.
- *DN: This is a process issue, not a test case requirement*

5) Establish and maintain a roadmap for the program

- basic platform verification

- well-established features, i.e. features that have become broadly supported by current OPNFV distros)
- *DN: This is a process issue, not a requirement for an individual test case*

6) Move tests upstream asap

- as soon as a home community is found, focus on upstreaming all tests

- I'm not sure how this will fit with a logo program, because in those programs you want to keep your tests stable and usually under your control, this seems to be more of a principle for the general test projects.

- Upstream test cases that we use are always version referenced so the stability is tied the speed of development in that community. The issue is less of stability but more of the compatibility of different versions of different components in the reference platform. I think this may be one of the differences between traditional testing scheme and open source.

- By the way, I think the integrated tests (inter-components) could reside in OPNFV, uniqueu features are another example, we may also be the natural home for nfv use cases etc.

- one more point: upstream testing is sometimes developer-oriented, not always focused on compliance. in such cases, we need a way of either enhancing the upstream test suite or coming up a methodoly in dovetail.
- *DN: Requirement added to underline that upstream test suites are a baseline, but I do not think that we can specify in test case requirements that the test has been proposed to the relevant upstream*