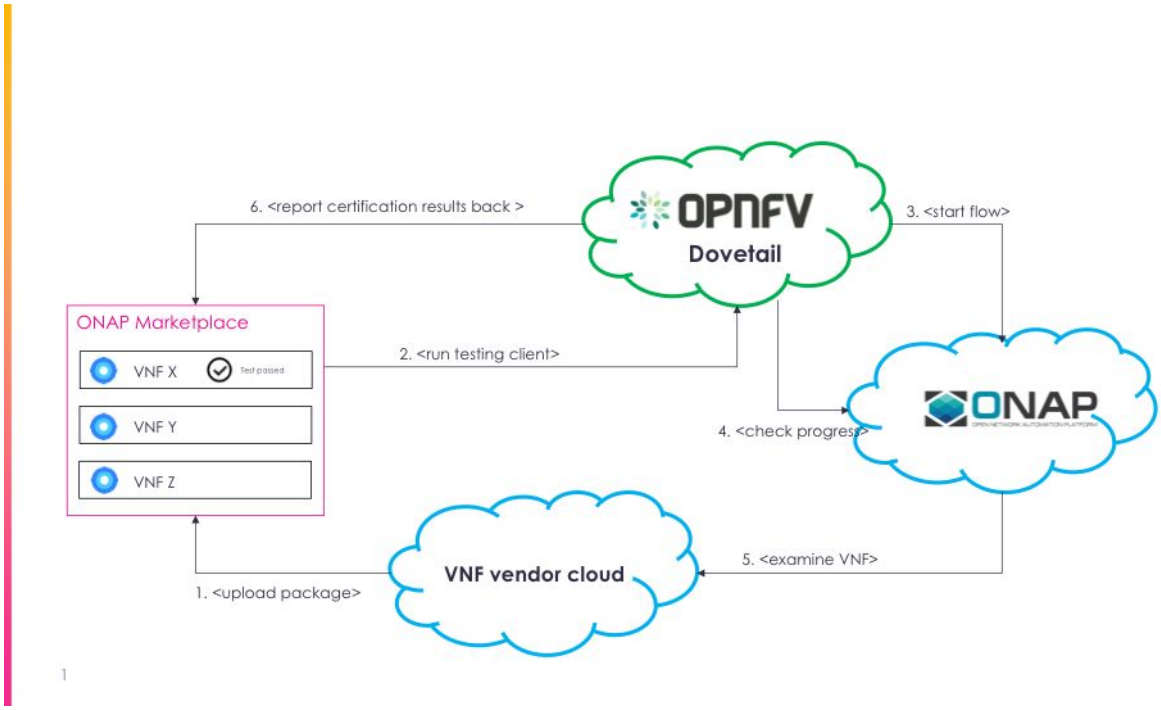
Dovetail

Dovetail Architecture Diagram Integration with VNF :
https://onap.readthedocs.io/en/latest/submodules/vnfsdk/model.git/docs/files/Dovetail.html



https://wiki.opnfv.org/display/dovetail/How+to+find+source+code+of+test+cases

Dovetail TEST CASES CODE LINK Area wise distribution :

For Dovetail test areas and test cases from Tempest, the links to them could be found by the following steps:

1. For each Dovetail test area, identify the related Tempest source code package. Specifically,
    a. the test area "VIM Operations on Compute" corresponds to https://github.com/openstack/tempest/tree/12.2.0/tempest/api/compute
    b. the test area "VIM Operations on Network" corresponds to https://github.com/openstack/tempest/tree/12.2.0/tempest/api/network
    c. the test area "VIM Operations on Volume" corresponds to https://github.com/openstack/tempest/tree/12.2.0/tempest/api/volume
    d. the test area "VIM Operations on Image" corresponds to https://github.com/openstack/tempest/tree/12.2.0/tempest/api/image
    e. the test area "VIM Operations on Identity" corresponds to https://github.com/openstack/tempest/tree/12.2.0/tempest/api/identity

2. For each test case in the above mentioned test areas, read its description and recognize the related modules in the Tempest package identified by the first step. For example, the links to the test case "Images member CRUD ops" in the test area "VIM Operations on Image" are as follows:
    a. https://github.com/openstack/tempest/blob/12.2.0/tempest/api/image/v2/test_images_member.py
    b. https://github.com/openstack/tempest/blob/12.2.0/tempest/api/image/v2/test_images_member_negative.py
    c. https://github.com/openstack/tempest/blob/12.2.0/tempest/api/image/v1/test_image_members.py
    d. https://github.com/openstack/tempest/blob/12.2.0/tempest/api/image/v1/test_image_members_negative.py

For other Dovetail test areas and test cases, the links to them are listed as follows:

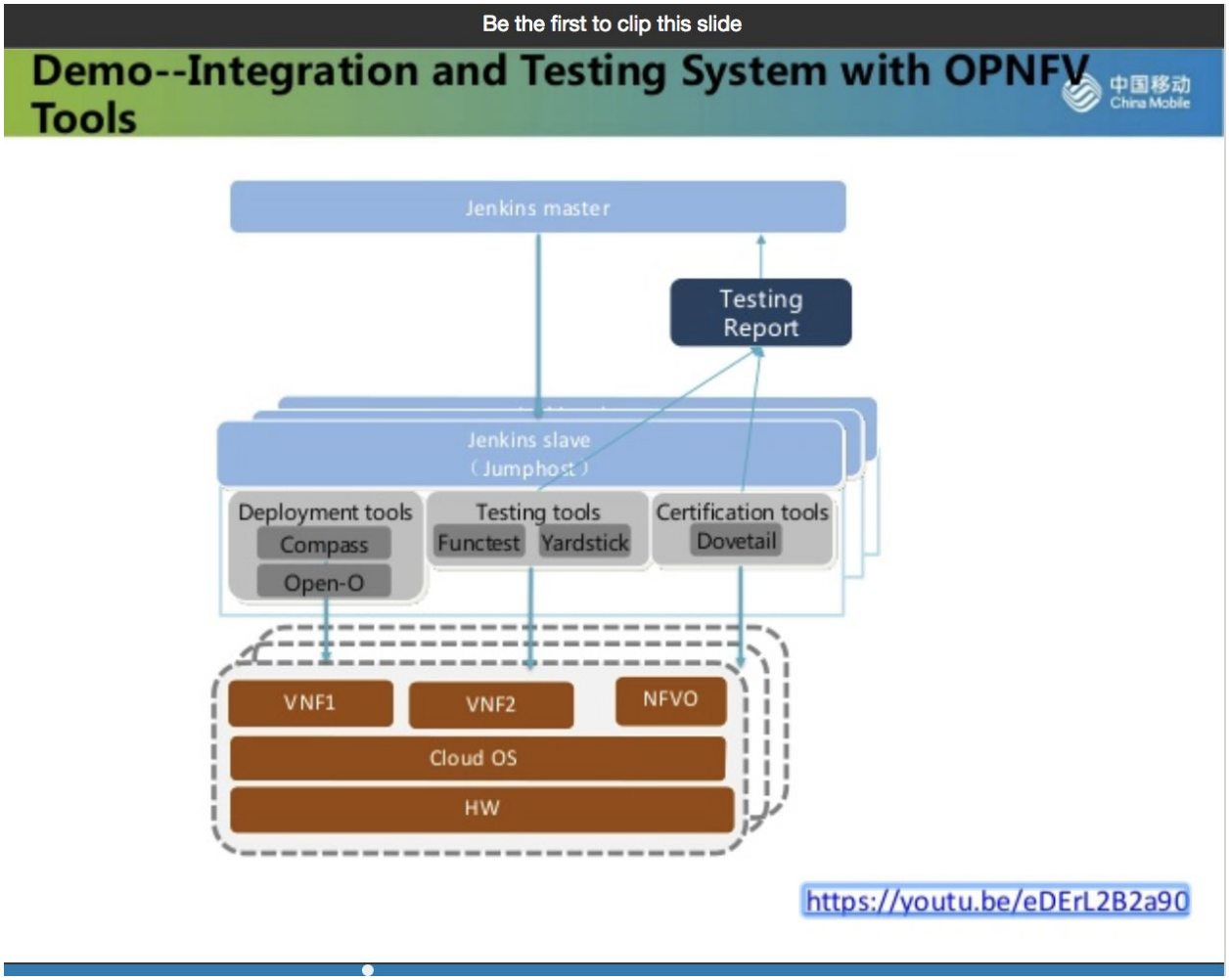
1. **The source code for the test area "NFVI" mainly consists of three packages:**
    a. the test case "vPing" corresponds to https://github.com/opnfv/functest/blob/colorado.1.0/testcases/OpenStack/vPing/vping.py
    b. the test cases named with prefix "OPNFV_YARDSTICK_" correspond to https://github.com/opnfv/yardstick/tree/colorado.1.0/tests/opnfv/test_cases
    c. the others correspond to https://github.com/openstack/tempest/tree/12.2.0/tempest/scenario
2. The link to the test cases named with prefix "OPNFV_YARDSTICK_" in the test area "HA" is https://github.com/opnfv/yardstick/tree/colorado.1.0/tests/opnfv/test_cases
3. The links to the test area "IPv6" are as follows:
    a. https://github.com/openstack/tempest/blob/12.2.0/tempest/api/network/test_dhcp_ipv6.py
    b. https://github.com/openstack/tempest/blob/12.2.0/tempest/api/network/test_networks.py
    c. https://github.com/openstack/tempest/blob/12.2.0/tempest/scenario/test_network_v6.py
    d. https://github.com/opnfv/yardstick/blob/colorado.1.0/tests/opnfv/test_cases/opnfv_yardstick_tc027.yaml
4. The link to the test area "VPN" is https://github.com/opnfv/sdnvpn/tree/colorado.1.0/test/functest
5. The link to the test area "Doctor" is https://github.com/opnfv/doctor/blob/colorado.1.0/tests/run.sh
6. The link to the test area "KVM" is https://github.com/opnfv/kvmfornfv/blob/colorado.1.0/tests/cyclictest.sh
7. The link to the test area "Parser" is https://github.com/opnfv/parser/blob/colorado.1.0/tests/functest_run.sh
8. The link to the test cases named with prefix "opnfv_yardstick_" in the test area "virtual Traffic classifier" is https://github.com/opnfv/yardstick/tree/colorado.1.0/tests/opnfv/test_cases

9. The link to the test area "Copper" is https://github.com/opnfv/copper/tree/colorado.1.0/tests
10. The link to the test area "Promise" is https://github.com/opnfv/promise/blob/colorado.1.0/source/test/promise-intents.coffee
11. The link to the test area "Multisite" is https://github.com/openstack/kingbird/tree/0.2.1/kingbird/tests/tempest/scenario/quota_management/client_tests
12. The links to the test area "ODL" are as follows:
    a. the test case "restconf modules" corresponds to https://github.com/opendaylight/integration-test/tree/release/beryllium-sr3/csit/suites/integration/basic
    b. the test cases "neutron networks/subnets/ports" correspond to https://github.com/opendaylight/integration-test/tree/release/beryllium-sr3/csit/suites/openstack/neutron
    c. the others correspond to https://github.com/opnfv/functest/tree/colorado.1.0/testcases/Controllers/ODL/custom_tests/neutron
13. The link to the test area "ONOS" is https://github.com/wuwenbin2/OnosSystemTest/tree/master/TestON/tests
14. The link to the test area "Open source VNF running on NFVI" is https://github.com/opnfv/functest/blob/colorado.1.0/testcases/vnf/vIMS/vIMS.py

Dovetail Code Base: https://github.com/opnfv/dovetail

Hogew Testing is Done in OPNFV :

https://www.youtube.com/watch?v



=eDErL2B2a90&feature=youtu.be

Dovetail Test Suite

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.

How to install Dovetail Test Suite on JumpHost :
https://wiki.opnfv.org/display/dovetail/Dovetail+Testing+Guide

Dovetail Test Runner :
https://github.com/opnfv/dovetail/blob/master/dovetail/test_runner.py

VNF Integration with Dovetail Video :
https://wiki.onap.org/display/DW/VNFTEST+integration+with+DOVETAIL?preview=/28377754/28377756/onap-opnfv-collaboration-demo-21_march_2018.mp4

################## DOVETAIL ########################

Jio Generic Testing Requirements doc:

(RUN-TIME TESTING):

A. Operate phase testing:

1. Vendor VNF Scalability testing
    a. VNF Scaling-Out testing by instantiating new VNFCs
    b. VNF Scaling-Out testing by allocating additional NFVI resources

2. Vendor VNF Auto - Healing testing
    a. VNF failover testing
    b. VNF migration testing

B. Maintain phase testing:

1. VNF service termination testing
2. VNF service suspend and resumption
3. VNF backup and restore testing

DOVETAIL INTEGRATION WITH VNFSDK:
1. Lifecycle
2. Onboarding
3. Common
4. Steps - validation tests using AAI

######################## VNF PACKAGE ################################

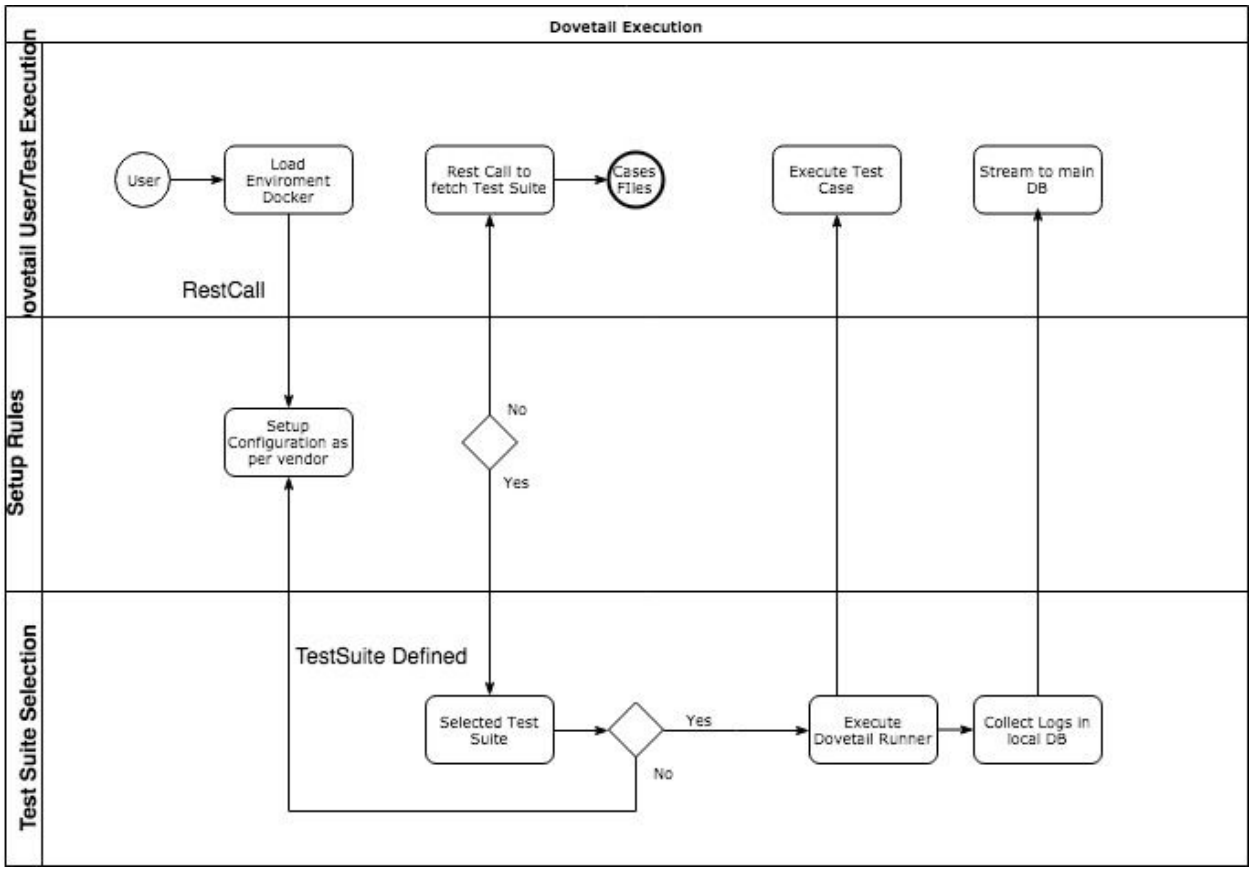
1. VNF Package comes from the marketplace
APIs listed in the vnfmarket-be / vnf-sdk-marketplace / src / main / java / org / onap / vnfsdk / marketplace / entity / request and response

####################### Open - O / ECOMP documentation ######################

Look into the image and doc posted on slack yesterday

###################### Robot test framework ############################
Health checks
Also look into functional tests APIs (tutotrials - use case specific)
vnfsdk/functest.git / vnf-sdk-function-test / src / main / java / org / onap / vnfsdk / functest

Dovetail Workflow:



Test Suite and Setup Information in Doc Link :

https://docs.google.com/document/d/1e6J2tuwwDMl4ueqHl4iHvfiu2qyPHP1nRVBOVtXl9tQ/edit?usp=sharing

Dovetail Test Execution Workflow