Automatic Integration and Testing using OPNFV tools in China Mobile

Fu Qiao, China Mobile, fuqiao@chinamobile.com
Gao Liang, Huawei, jean.gaoliang@huawei.com
Index

- NFV Integration and Testing, scenarios and requirements
- Automatic integration and testing framework
- Demo
**Network Function Virtualization** has been widely experimented and deployed around the world. Operators are continuously moving traditional Network services to the NFV Cloud, in order to reduce its cost and accelerate service launching. Through the years, NFV has developed from concept and framework to field trial and real deployment.
China Mobile is building its NFV Experiment Network in 4 Provinces.

Multiple DC sites are included in the network in each province.

Multiple NFV platforms and VNFs are tested.

Due to limited resource in provinces, integration and testing are first done in Labs in Beijing, and are further deployed in each Province.
China Mobile structures its NFV Cloud into TICs (Telecom Integrated Cloud)

TIC is a standard unit, with unified hardware and software. Multiple VNF services are deployed in TICs

In the Experiment network, core TIC and Regional TIC are deployed in each province.
TIC: Standard Unit for NFV Deployment

TIC is a standard unit for NFV Deployment
- Limited number of design template, including NFVI, VIM, and NFVO
- Unified Hardware model
- Standard network design

**Design Template**
4 TIC design template to carry VNF services of different kinds, including:
- Template for Control plane services, stressing on compute capability, e.g. vIMS
- Template for Data plane services, stressing on data forwarding capability, e.g. S/P-GW
- Template for Edge services, stressing on low cost and light weight, e.g. vCPE
- Template for Storage services, stressing on storage capability, e.g. vCDN

**Standard Network**
- Strict network plane partition, including Management net, service net, and storage net.
- SG design

**Unified Hardware**
- Unified hardware design with limited number of models suitable for different category of services.
HW configuration
- HW deployment and configuration according to standard Hardware requirement

Platform Deployment
- TIC Platform deployment according to standard templates

Platform Testing
- Functional, performance and availability testing of TIC Platform according template requirements.

VNF Deployment
- VNFM, VNF, NFVO Deployment and integration

VNF Testing
- Functional, performance and availability testing according to service requirements
On Going Work in China Mobile’s Lab

- Build Jenkins system for automation
- Using OPNFV open-source test tools for Infrastructure testing.
- Using OPNFV installers for automatic deployment
- Using proprietary testing tools for VNF service testing.
Automatic Integration & Testing System

System Design and Configuration
- Overall Design
- Detailed Design
- Hardware Configuration

Platform Deployment & Testing
- VIM+NFVI Deployment with automatic installer
- Functional, performance, and reliability testing using automatic test tools

Orchestrator, VNFM Deployment & Testing
- Orchestrator and VNFM Deployment with automatic installer
- Interface testing

VNF Onboarding & Testing
- VNF onboarding with Orchestrator and VNFM
- VNF service testing

Automatic Integration & Testing Procedure

Jenkins
- Download installer job
- Download Infrastructure testing job
- Download service onboarding testing job

NFVO
VNF
VNF
VNF
VNFM
Hypervisor
VIM
Hardware
Introduction of Compass

Deploy to BareMetal
- Auto-discovery of Hardware
- Support various servers, switches, storages

Package Installer
- Openstack deployment with various flavors
- VNF (vIMS, epc, etc)
- Extensible - easy to add new applications

OS-Installer
- Cobbler(PXE)
- OpenStack Ironic

Application management
- Application monitor and logging
OPNFV has been devoted in NFV testing for a long term. It currently has comprehensive testing tools, including functional testing, performance testing, high availability testing and etc. It also provides a whole set of test framework, with each different test tools as containers, making them easier to coordinate.
## Functest Test Case Overview

<table>
<thead>
<tr>
<th>OpenStack</th>
<th>OPNFV Feature Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>vPing test case</td>
<td>Copper test case</td>
</tr>
<tr>
<td>Create machines and verify connectivity</td>
<td>Doctor test case</td>
</tr>
<tr>
<td>Rally bench tests</td>
<td>Domino test case</td>
</tr>
<tr>
<td>Benchmark the OpenStack deployment</td>
<td>Multisite test case</td>
</tr>
<tr>
<td>Tempest test</td>
<td>Promise test case</td>
</tr>
<tr>
<td>OpenStack native tests (100+ smoke-tests)</td>
<td>SDNVPN test case</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SDN</th>
<th>VNF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODL test case</td>
<td>vIMS test case</td>
</tr>
<tr>
<td>Robot framework, ODL functional testing</td>
<td>Clearwater vIMS Test Cases With Cloudify</td>
</tr>
<tr>
<td>ONOS test case</td>
<td></td>
</tr>
<tr>
<td>TestON, ONOS system testing</td>
<td></td>
</tr>
</tbody>
</table>

- OpenStack
  - vPing test case: Create machines and verify connectivity
  - Rally bench tests: Benchmark the OpenStack deployment
  - Tempest test: OpenStack native tests (100+ smoke-tests)

- SDN
  - ODL test case: Robot framework, ODL functional testing
  - ONOS test case: TestON, ONOS system testing

- OPNFV Feature Projects
  - Copper test case
  - Doctor test case
  - Domino test case
  - Multisite test case
  - Promise test case
  - SDNVPN test case

- VNF
  - vIMS test case: Clearwater vIMS Test Cases With Cloudify
• Support RESTful API, can be easily integrated
• Support plug-in, can integrate other tools
• Support system level configuration
• Support test case control
• Support test result distribution and visualization
• Test case decomposed with framework
Introduction of Dovetail

**Dovetail** is the project that delivers open source software toolchain, documentation, and test suite for compliance verification program (CVP) in OPNFV community. It leverages upstream testing projects to make sure that a NFVI platform is in compliance with OPNFV requirements.

![Diagram of Dovetail setup](image)

- **A vendor’s domain**
  - A host or VM or container to run dovetail
  - The internal network
  - SUT

- **The external network** with HTTPS open
  - Use e.g. LF ID to login & check results

- **Web server and DB** hosted and administered by e.g. OPNFV or hired help.
  - e.g. https://cvp.opnfv.org

- **A reviewer**
  - (client)
  - (server)

(A vendor can also do dry-runs by installing a local server. * May also add a local file dump for simplicity.)
Demo -- Integration and Testing System with OPNFV Tools

Jenkins master

Testing Report

Jenkins slave (Jumphost)

Deployment tools
  - Compass
  - Open-O

Testing tools
  - Functest
  - Yardstick

Certification tools
  - Dovetail

VNF1
VNF2
NFVO
Cloud OS
HW

https://youtu.be/eDERL2B2a90