

**1<sup>st</sup> ETSI NFV Plugtests**  
**Madrid, Spain**  
**23rd January – 3<sup>rd</sup> February**

---



---

**Keywords**

Testing, Interoperability, NFV, MANO, VNF, VIM

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-préfecture de Grasse (06) N° 7803/88

---

**Important notice**

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services: [http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI. The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2017.  
All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **GSM®** and the GSM logo are Trade Marks registered and owned by the GSM Association.

# Contents

Contents .....	3
Executive summary .....	5
1 Introduction .....	6
2 References .....	7
3 Abbreviations .....	8
4 Technical and Project Management .....	9
4.1 Scope .....	9
4.2 Timeline .....	9
4.2.1 Documentation .....	10
4.2.2 Remote integration & pre-testing .....	10
4.2.3 Plugtests .....	11
4.3 Tools .....	11
4.3.1 Plugtests Wiki .....	11
4.3.2 Test Session Scheduler .....	12
4.3.3 Test Reporting Tool .....	12
5 Functions Under Test .....	14
5.1 VNFs .....	14
5.2 MANOs .....	14
5.3 VIM&NFVI .....	15
5.4 Test Tools .....	15
6 Test Infrastructure .....	16
6.1 Remote Test Infrastructure .....	16
6.2 Local Test Infrastructure .....	17
7 Test Procedures .....	18
7.1 Remote Integration & Pre-testing Procedure .....	18
7.2 Interoperability Testing Procedure .....	19
8 Test Plan Overview .....	24
8.1 Introduction .....	24
8.2 Test Groups .....	24
8.2.1 Setup .....	24
8.2.2 Network Service Life Cycle Management .....	24
8.2.2.1 Instantiate .....	24
8.2.2.2 Scale .....	25
8.2.2.3 Scale VNF .....	26
8.2.2.4 Update .....	27
8.2.2.5 Terminate .....	28
8.2.3 Teardown .....	28
9 Interoperability Results .....	29
9.1 Overall Results .....	29
9.2 Results per Test Group .....	30
9.3 Results per Test Case .....	30
9.3.1 Test Group: Setup & Instantiation .....	31
9.3.2 Test Group: Scale .....	31
9.3.3 Test Group: Scale VNF .....	32
9.3.4 Test Group: Update .....	33
9.3.5 Test Group: Terminate & Teardown .....	33
10 Plugtests Outcome .....	34
10.1 Feedback on the Test Plan .....	34
10.1.1 NS Update – Stop / Re-start VNF .....	34
10.1.2 SFC Testing .....	35

10.2	Feedback on NFV Specifications.....	36
10.2.1	NFV-IFA013. Update NS (Operate VNF) .....	36
10.2.2	NFV-IFA007/NFV-IFA008 vs NFV-SWA001: VNF Instance States .....	36
10.2.3	NFV-IFA008 Ve-VNFM Interfaces .....	36
10.3	Feedback on IOP Issues .....	37
10.3.1	VNF management network exposure .....	37
10.4	Other Feedback .....	37
10.4.1	NFVI Dimensioning and Configuration .....	37
Annex A	VNF and NS Examples .....	38
A.1	Example NS#1: Testing an endpoint VNF .....	38
A.1.1	Example VNF#11: Endpoint VNF .....	39
A.1.2	Example VNF#21: Generator 1 port .....	40
A.2	Example NS #2: Testing a middle point VNF .....	41
A.2.1	Example VNF#12: Middle point VNF .....	42
A.2.2	Example VNF#22: Generator 2 ports.....	43
History	.....	44

---

## Executive summary

The 1<sup>st</sup> NFV Plugtests was organised by ETSI Centre for Testing and Interoperability, and hosted by 5TONIC Laboratory in Leganes, near Madrid, Spain, from 23 January to 03 February 2017.

A pre-testing and remote integration phase was launched in November 2016. 29 remote labs were connected to the ETSI Plugtests network to verify interconnection and compatibility among the different Virtualized Network Functions (VNFs), Management and Orchestration (MANO) solutions and NFV platforms participating in the Plugtests.

During the two-week intense testing phase at 5TONIC, multi-vendor interoperability test sessions were run focusing on validating ETSI NFV Release 2 end-to-end capabilities including management of descriptors and software images, as well as life cycle management of network services and virtual network functions.

The test plan development was driven by ETSI's Centre for Testing and Interoperability, building on the methodology defined by the ETSI NFV testing working group.

During the Plugtests, 35 commercial and open source implementations participated in 160 interoperability test sessions in which the System under Test was made of different combinations of 15 virtual network functions, 9 management and orchestration solutions and 11 NFV platforms. Over 31 organisations and 160 engineers were involved in the preparation of this two week event, forming an engaged and diverse new community of NFV implementers.

Key NFV Open Source projects, like ETSI OSM, Open Baton, OPEN-O and OPNFV were also present at the Plugtests and participated in the test sessions. This first ETSI NFV Plugtests was a unique opportunity to stimulate synergy and alignment across the NFV ecosystem.

The overall results of this first NFV Plugtests show high rates of test execution and interoperability for features like Network Service on-boarding, instantiation and termination and very encouraging initial results for complex operations like scaling and Network Service updates. The test plan, overall results and lessons learnt during the Plugtests are fed back to ETSI NFV Industry Specification Group

---

# 1 Introduction

This Plugtests aimed at verifying early interoperability between different implementations of the main components of the NFV Architectural Framework, which included:

- Virtual Network Functions (VNF)
- Management and Orchestration (MANO) solutions, providing pre-integrated NFV Orchestrator (NFVO) and VNF Manager (VNFM) functionality
- NFV Platforms providing pre-integrated NFV infrastructure (NFVI) and Virtual Infrastructure Manager (VIM) functionality

Test and support VNFs were used to build the reference Network Services (NS) required to validate the proper behaviour of the Systems Under Test.

Most of the NFV Platforms were connected remotely to the Plugtests network. All the MANO solutions were deployed and run remotely from participant's labs. The VNF and NS Packages were made available in a local Plugtests repository, and uploaded to the different NFV platforms during the pre-testing phase.

Some test and support VNFs were deployed locally in the 5TONIC laboratory, located in the IMDEA Networks facilities.

All the information required to organise and manage the 1<sup>st</sup> NFV Plugtests was compiled and shared with participants in a dedicated private WIKI put in place by ETSI. Participants were provided with credentials that allowed them to access and update their details. All the information presented in this document has been extracted from the 1<sup>st</sup> NFV Plugtests wiki: <https://wiki.plugtests.net/1st-NFV-Plugtests> (login required).

---

## 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

- [NFV002] ETSI GS NFV 002: "Network Functions Virtualisation (NFV); Architectural Framework".
- [NFV003] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".
- [IFA005] ETSI GS NFV-IFA 005: "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".
- [IFA006] ETSI GS NFV-IFA 006: "Network Functions Virtualisation (NFV); Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".
- [IFA007] ETSI GS NFV-IFA 007: "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".
- [IFA008] ETSI GS NFV-IFA 008: "Network Functions Virtualisation (NFV); Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".
- [IFA010] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV); Management and Orchestration; Functional requirements specification".
- [IFA013] ETSI GS NFV-IFA 013: "Network Functions Virtualisation (NFV); Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification".
- [TST002] ETSI GS NFV-TST 002: "Network Functions Virtualisation (NFV); Testing Methodology; Report on NFV Interoperability Testing Methodology"
- [SWA001] ETSI GS NFV-SWA 001: "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture
- [1NFVPLU-TP] 1<sup>st</sup> ETSI NFV Plugtests Test Plan:  
[https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/1st\\_ETSI\\_NFV\\_Plugtests\\_Test\\_Plan\\_v1.0.0.pdf](https://portal.etsi.org/Portals/0/TBpages/CTI/Docs/1st_ETSI_NFV_Plugtests_Test_Plan_v1.0.0.pdf)

---

## 3 Abbreviations

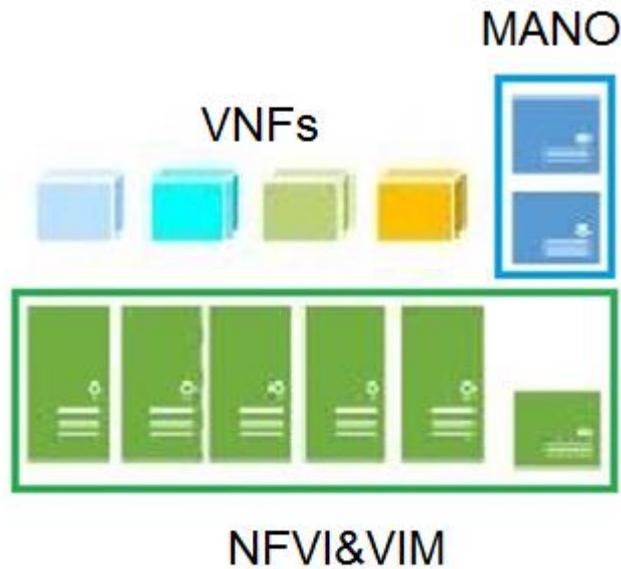
For the purposes of the present document, the terms and definitions given in [NFV003] and [TST002] apply.

## 4 Technical and Project Management

### 4.1 Scope

The main goal of the first NFV Plugtests was the validation of ETSI NFV Release 2 end-to-end capabilities including management of descriptors and software images, as well as life cycle management of network services, virtual network functions as well as virtual resources.

The Systems Under Test (SUTs) were composed of combinations of several Functions Under Test (FUT), each of them provided by different participants.



**Figure 1. System Under Test**

In the scope of this Plugtests, the Systems Under Test (SUT) included the following Functions Under Test (FUTs):

- One NFV Platform providing pre-integrated NFV infrastructure (NFVI) and Virtual Infrastructure Manager (VIM) functionality
- One Management and Orchestration (MANO) solution, providing pre-integrated NFV Orchestrator (NFVO) and VNF Manager (VNFM) functionality
- One or Several Virtual Network Functions (VNF)

Due to the early stages of the NFV standards, conformance to NFV Interfaces specifications was not enforced during this “early” Plugtests and interoperability among the different Functions Under Test was achieved through the exposure of open APIs or plugins and a phase of remote integration.

### 4.2 Timeline

The 1<sup>st</sup> NFV Plugtests preparation was run through different phases as described in the figure below.

JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB
Registration			Documentation	Remote int. & pre-testing		Plugtests	Report

**Figure 2. Plugtests timeline**

Registration to Plugtests was open from July to September 2016 to any organisation willing to participate with a Function Under Test or Test Function. Due to space limitations in the hosting laboratory, on-site participation was limited to 64 people at a time. Participating companies were requested to restrict on-site participation to a maximum of 2 people per implementation at a time. Additional remote participation (i.e. back office support) was possible and supported with electronic tools, see clause 4.3. A total of 160 people were involved in the Plugtests either locally or remotely.

The following clauses describe the different phases of the Plugtests preparation. It is worth noting that since the start of the documentation phase until the 2 weeks of face to face Plugtests, weekly conf-calls were run among organisers and participants to discuss and track the progress, anticipate and solve technical issues, review the test plan, etc..

## 4.2.1 Documentation

Once registration to the Plugtests was closed, the following documentation activities were launched in parallel:

### 1) FUT Documentation

Participants documented their FUTs, by filling in a form compiling the Interoperability Features Statement (IFS) and Technical Questions (TQ) concerning their implementations. The final IFS Templates for each type of FUT was appended to the Plugtests Test Plan,

Participants providing VNFs complemented their documentation with diagrams and resources requirements. Some example VNFs and NSs were made available by the Plugtests organisers to facilitate the documentation of participating VNFs. See VNF and NS documentation examples in Annex A.

Participants providing MANO solutions developed and made available descriptor samples for the VNF and NS examples and supported the VNF providers in the creation of their own VNF and NS Descriptors.

Participants providing NFV Platforms created and documented tenants and credentials for each participating MANO solution, and exposed and documented their North Bound Interface (NBI) of their VIM.

All the information described above was made available in the Plugtests WIKI, so that it could be easily maintained and consumed by participants.

### 2) Test Plan Development

The Test Plan development was led by ETSI Centre for Testing and Interoperability following the methodology defined by ETSI NFV TST WG in TST002. The Test Plan was scoped around ETSI NFV Release 2 capabilities and concentrated on the features supported by the implementations attending the Plugtests. The supported features were compiled thanks to the IFS filled in by participants.

The Test Plan was developed and consolidated in an iterative way, taking into account input and feedback received from ETSI NFV TST WG, supporting Open Source Projects and Plugtests participants. See details in clause 8.

## 4.2.2 Remote integration & pre-testing

Starting in November 2016, participants connected their implementations remotely to the Plugtests infrastructure, known as HIVE: Hub for Interoperability and Validation at ETSI.

During this phase, up to 29 remote labs connected to HIVE and each of them was allocated a dedicated network. The interconnection of remote labs allowed running integration and pre-testing tasks remotely among any combination of participating FUTs, in order to ensure an efficient use of the face to face Plugtests time and smoother Interoperability Test Sessions.

A VPN connection to HIVE was mandatory for participants providing NFV Platforms and MANO Solutions, and highly recommended for participants providing VNFs, for trouble shooting and infrastructure access purposes.

Additional details on the remote test infrastructure, remote integration and pre-testing procedures are provided in Clauses 6 and 7.

During this phase, weekly conf-calls were run among organisers and participants to synchronise, track progress and get ready for the on-site phase.

## 4.2.3 Plugtests

From 23<sup>rd</sup> of January to the 3<sup>rd</sup> of February, participants sent representatives to the host Lab to collaboratively run the Interoperability Test Sessions.

These 2 weeks on-site were organised as follows:

1ST NFV PLUGTESTS Agenda (23 JAN - 03 FEB 2017)												
Time	Monday 23	Tuesday 24	Wednesday 25	Thursday 26	Friday 27	Saturday 28	Sunday 29	Monday 30	Tuesday 31	Wednesday 1	Thursday 2	Friday 3
08:30 10:00			WELCOME & WARM UP	WARM UP	WARM UP			WARM UP				
10:00 13:00	LOCAL SW/HW INSTALLATION & PRE-TESTING	LOCAL SW/HW INSTALLATION & PRE-TESTING	TEST SESSIONS	TEST SESSIONS	TEST SESSIONS			TEST SESSIONS				
13:00 14:30	LUNCH BREAK	LUNCH BREAK	LUNCH BREAK	LUNCH BREAK	LUNCH BREAK			LUNCH BREAK				
14:30 17:30	LOCAL SW/HW INSTALLATION & PRE-TESTING	LOCAL SW/HW INSTALLATION & PRE-TESTING	TEST SESSIONS	TEST SESSIONS	TEST SESSIONS			TEST SESSIONS	TEST SESSIONS	TEST SESSIONS	TEST SESSIONS	TEAR DOWN
17:30 18:30		BRIEFING	WRAP UP	WRAP UP	WRAP UP			WRAP UP	WRAP UP	WRAP UP	WRAP UP	

**Figure 3. Plugtests on-site HL schedule**

The 2 first days were dedicated to local installation and pre-testing continuation, this time also including local implementations. A number of FUTs were installed and connected locally to the HIVE infrastructure, as well as some test and support functions.

The following 7 1/2 days were dedicated to on-site interoperability test sessions involving all the participating FUTs organised in several parallel tracks, see details in Clause 4.3.2.

## 4.3 Tools

### 4.3.1 Plugtests Wiki

The Plugtests Wiki was the main entry point for all the information concerning the NFV Plugtests, from logistics aspects to testing procedures. Access to the WIKI was restricted to participating companies.

The main technical information provided in the wiki was organised as follows:

- **Event Information** – Logistics aspects of the Plugtests
- **Network Information** - HIVE connection request tool, and remote connections status overview
- **Functions Under Test** - Participating FUTs overview, IFS/TQ forms, IFS/TQ responses overview, FUT dedicated pages.
- **Pre-Testing** - Remote integration and pre-testing progress tracking matrix, remote integration and pre-testing procedures
- **VNF & NS Examples** - Diagram and Requirements documentation examples for sample VNFs / NS (end-point, middle point...)
- **Testing Information** - Access to the Test Plan, including Test Suite Structure, SUT Configurations and Test Descriptions
- **Schedule & Reporting** - Daily schedule for the on-site phase, access and documentation of the Test Reporting Tool
- **Conf-Calls** - Calendar, logistics, agendas and minutes of the weekly conf-calls run during the remote integration and pre-testing phase.
- **Wrap-up Sessions** - Agenda and minutes of the daily wrap up meetings run during the on-site phase.

In addition, an embedded IRC allowed participants to communicate during the pre-testing phase and Test Sessions, and include their remote colleagues (back-office support) in the discussions.

### 4.3.2 Test Session Scheduler

The Test Session Scheduler allowed the Plugtests organisers to produce a daily schedule during the on-site phase. This tool has the following objectives:

- maximise the number of test sessions
- balance the amount of test sessions among participants
- take into account supported features of the participating FUTs
- minimise the number of participants not involved in a test session anytime.

The picture below shows a partial view a daily schedule. Each yellow box corresponds to a specific Test Session including 1 VNF, 1 MANO solution and 1 VIM&NFVI. For each of these sessions a Test Session Report was recorded (see next clause). An average of 22 Test Sessions were run every day during the Plugtests.

09:00-10:00											
ADVA - Easanta VNF: Lantier - NPTV4 Futur - Futura	Ericsson - Cloud Manager VNF: Telecom Inbi - NPA-6 CI	Ericsson - Cloud Manager VNF: Telecom Madsen - Madsen	Franklin POCUS - Opande VNF: Lantier - NPTV4 Spart - CloudStream	SP Emporia - NPT Director VNF: Lantier - NPTV4 Auton - ManuCar-Orphe	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Rahma - Alena	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 All Networks - T-Trade	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Santana - PPS	RUI - RUI.vms VNF: Lantier - NPTV4 Sana - SSC - EMS	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Santana - PPS	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Santana - PPS	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Santana - PPS
10:00-11:00											
ADVA - Easanta VNF: Lantier - NPTV4 Rahma - Alena	ADVA - Easanta VNF: Lantier - NPTV4 Santana - PPS	Ericsson - Cloud Manager VNF: Telecom Madsen - TA	Ericsson - Cloud Manager VNF: Telecom Cyant - POCF	Franklin POCUS - Opande VNF: Lantier - NPTV4 Kajant - FinSAC	Franklin POCUS - Opande VNF: Lantier - NPTV4 All Networks - T-Trade	SP Emporia - NPT Director VNF: Lantier - NPTV4 Pai - Aho Networks - VNF	SP Emporia - NPT Director VNF: Lantier - NPTV4 Sana - SSC - EMS	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Auton - ManuCar-Orphe	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 EAST - NPT Test Automate	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Inbi - NPA-6 CI	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Santana - PPS
11:00-12:00											
ADVA - Easanta VNF: Lantier - NPTV4 Rahma - Alena	ADVA - Easanta VNF: Lantier - NPTV4 Santana - PPS	Ericsson - Cloud Manager VNF: Telecom Madsen - TA	Ericsson - Cloud Manager VNF: Telecom Cyant - POCF	Franklin POCUS - Opande VNF: Lantier - NPTV4 Kajant - FinSAC	Franklin POCUS - Opande VNF: Lantier - NPTV4 All Networks - T-Trade	SP Emporia - NPT Director VNF: Lantier - NPTV4 Pai - Aho Networks - VNF	SP Emporia - NPT Director VNF: Lantier - NPTV4 Sana - SSC - EMS	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Auton - ManuCar-Orphe	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 EAST - NPT Test Automate	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Inbi - NPA-6 CI	OSM - OSM.Rat.CORE VNF: Lantier - NPTV4 Santana - PPS

Figure 4. Daily Schedule example

### 4.3.3 Test Reporting Tool

The Test Reporting Tool guides participants through the Test Plan during the on-site Test Sessions, and allows them to create Test Session Reports compiling detailed results for the scheduled Test Sessions. It also allows reporting on Test Sessions organised on the fly among participants to complete or complement the planned testing (freestyle sessions).

Only the companies providing the FUTs for each specific Test Session have access to the Test Session Reports contents and specific results. All the companies providing the FUTs for a Test Session, i.e VNF provider, MANO provider and NFV Platform provider are required to verify and approve the reported results at the end of the session.

Another interesting feature of this tool is the ability to generate real-time stats (aggregated data) of the reported results, per test case, test group, test session or overall results. These stats are available to all participants and organisers and allow tracking the progress of the testing with different levels of granularity, which is extremely useful to analyse the results.

id	status	date	duration	area	config	participants	commands
2038	🟢	2017-01-25 14:30	180	Track 1	NFV	ADVA - Ensemble RedHat - Lenovo - NFVI+VIM Anritsu - MasterClaw vProbe	📄 🗑️ 🔄
2039	🟢	2017-01-25 10:00	180	Track 2	NFV	Ericsson - Cloud Manager Windriver - Titanium Italtel - NM-S CI	📄 🗑️ 🔄
2041	🟢	2017-01-25 10:00	180	New Area	NFV	OSM - OSM Ral ONE Windriver - Lenovo - NFVI+VIM Keynetic - FlowNAC	📄 🗑️ 🔄
2042	🟢	Freestyle			NFV	Rift.io - Rift.ware OPNFV - Ericsson - NFVI+VIM Spirent - CloudStress	📄 🗑️ 🔄
2045	🟢	2017-01-25 10:00	180	New Area	NFV	Fraunhofer FOKUS - OpenBaton OPNFV - Intel - NFVI+VIM Netrounds - vTA	📄 🗑️ 🔄
2046	🟢	2017-01-25 10:00	180	Track 5	NFV	OSM - OSM Ral ONE Windriver - Lenovo - NFVI+VIM Anritsu - MasterClaw vProbe	📄 🗑️ 🔄
2047	🟢	Freestyle			NFV	Cisco - NFVO RedHat - Lenovo - NFVI+VIM Openet - vPCRF	📄 🗑️ 🔄
2048	🟢	2017-01-25 14:30	180	New Area	NFV	OSM - OSM Ral ONE Windriver - Lenovo - NFVI+VIM Netrounds - vTA	📄 🗑️ 🔄
2049	🟢	2017-01-25 14:30	180	Track 5	NFV	OSM - OSM Ral ONE Windriver - Lenovo - NFVI+VIM EANTIC - NFV Test Automation Tool	📄 🗑️ 🔄
2051	🟢	2017-01-25 14:30	180	Track 7	NFV	Rift.io - Rift.ware OPNFV - Ericsson - NFVI+VIM Italtel - NM-S CI	📄 🗑️ 🔄
2052	🟢	2017-01-25 10:00	180	New Area	NFV	Ericsson - Cloud Manager Windriver - Titanium Sonus - SBC - EMS	📄 🗑️ 🔄
2053	🟢	Freestyle			NFV	Cisco - NFVO RedHat - Lenovo - NFVI+VIM F5 - vADC	📄 🗑️ 🔄
2054	🟢	2017-01-25 14:30	180	Track 2	NFV	Ericsson - Cloud Manager Windriver - Titanium Openet - vPCRF	📄 🗑️ 🔄
2056	🟢	Freestyle			NFV	Cisco - NFVO RedHat - Lenovo - NFVI+VIM Spirent - CloudStress	📄 🗑️ 🔄
2057	🟢	Freestyle			NFV	Cisco - NFVO RedHat - Lenovo - NFVI+VIM Keynetic - FlowNAC	📄 🗑️ 🔄
2058	🟢	2017-01-25 14:30	180	New Area	NFV	ADVA - Ensemble RedHat - Lenovo - NFVI+VIM Sonus - SBC - EMS	📄 🗑️ 🔄
2059	🟢	2017-01-25 10:00	180	Track 1	NFV	ADVA - Ensemble RedHat - Lenovo - NFVI+VIM	📄 🗑️ 🔄

Figure 5. Test Reporting Tool

## 5 Functions Under Test

The tables below summarise the different Functions Under Test provided by the Plugtests participants, and the location from where they were connecting to the remote test infrastructure and/or providing remote support:

### 5.1 VNFs

Organisation	Solution	Location	Type
A10 Networks	vThunder	USA/Germany	ADC, Firewall. CGN
Anritsu	MasterClaw vProbe	Denmark	Service Assurance/Probe
EANTC	NFV TA	Germany	Tester
F5	vADC	Spain	Load Balancer
Fortinet	Fortgate	France	Firewall
Italtel	NM-S CI	Italy	IMS
Keynetic	FlowNAC	Spain	Firewall
Mahindra Comviva	NGage	India	Enterprise Messaging
Netrounds	vTA	Sweden	Tester
Openet	Policy Manager	Ireland	PCRF
Palo Alto Networks	Virtual Firewall	USA	Firewall
Radware	Alteon	Israel	Probe/Load Balancer
Sandvine	PTS	Canada	Deep Packet Inspection
Sonus	SBC SWe	France/USA	Session Border Controller
Spirent	CloudStress, STC Virtual	USA	Tester

Table 1. VNFs Under Test

### 5.2 MANOs

Organisations	Solution	Location	Type
ADVA	Ensemble Orchestrator	NC, USA	Commercial
Cisco	NFVO	Sweden	Commercial
Ericsson	Cloud Manager	New Jersey, USA	Commercial
Fraunhofer FOKUS	Open Baton	Germany	Open Source
HPE	NFV Director	France/Spain	Commercial
Huawei	Open-O	Santa Clara, USA	Open Source
Openet	Weaver	Dublin	Commercial
RIFT.io	RIFT.ware	USA	Commercial
RIFT.io Canonical Telefonica	OSM	France	Open Source

Table 2. MANOs Under Test

### 5.3 VIM&NFVI

Organisations	Solution	Location	Type
Adva	Ensemble VIM	NC, USA	OpenStack Kilo/Mitaka
Canonical	Canonical OpenStack	Madrid, Spain	OpenStack Newton 5TONIC POD
Canonical Lenovo	Canonical OpenStack	NC, USA	OpenStack Newton Lenovo OCP 'OP@L' POD
Intel	OPNFV Colorado	Portland, USA	OpenStack Mitaka
Ericsson	OPNFV Colorado	Sweden	OpenStack Mitaka + ODL
Telefonica	OpenVIM (OSM)	Spain	OpenVIM + Floodlight/ODL OSM Remote Lab
Red Hat Lenovo	Red Hat Openstack Platform 9	NC, USA	OpenStack Mitaka Lenovo OCP 'OP@L' POD
VMware	vCloud NFV	Canada	vCloud + NSX
Wind River	Titanium Cloud	Santa Clara, USA	OpenStack Mitaka w/Regions + OF1.3 OSM Remote Lab
Wind River Lenovo	Titanium Cloud	NC, USA	OpenStack Mitaka + OF1.3 Lenovo OCP 'OP@L' POD
Wind River Intel	Titanium Cloud	Portland, USA	OpenStack Mitaka + OF1.3 Intel POD

Table 3. VIM&amp;NFVIs Under Test

### 5.4 Test Tools

In addition of the Functions Under Test, some participants made their Test Functions and Tools available to support the Test Sessions. Use of these tools was optional.

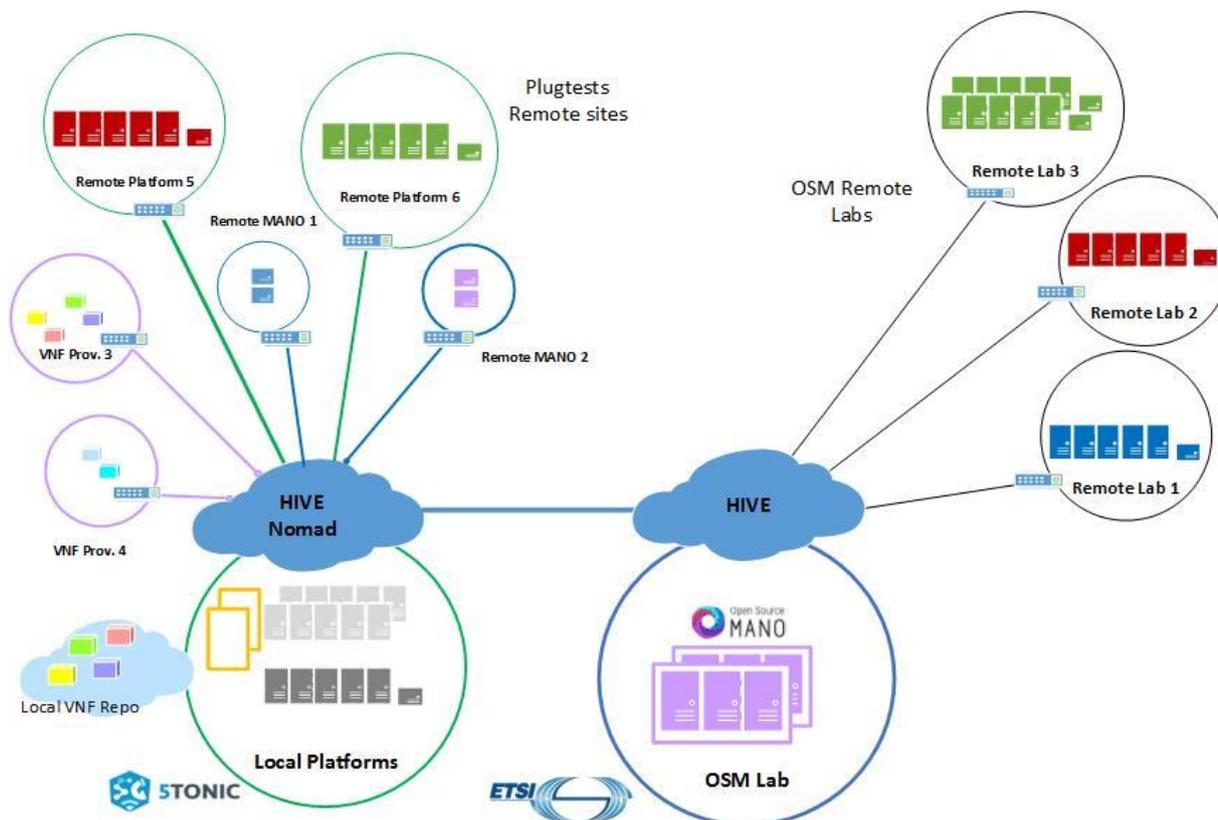
Organisation	Tool Description
EANTC	Test Automation Tool
Ixia	Traffic Generator, Protocol Emulator, Performance Test Tool
Netrounds	Test and Measurement Tool
Spirent	Traffic Generator, Analyser, Capture, Protocol Emulator, Performance Test Tool. Synthetic VNF Workload Generator for CPU, Memory, Storage and Network

Table 4. Test Tools

## 6 Test Infrastructure

### 6.1 Remote Test Infrastructure

The remote integration and pre-testing phase was enabled by the following setup:



**Figure 6. Remote Test Infrastructure**

The remote test infrastructure leveraged ETSI’s HIVE (Hub for Interoperability and Validation and ETSI) and the OSM Remote Labs network as follows:

The OSM Remote Labs network includes several of OSM instances running locally at ETSI and connected through HIVE to a number of Remote Labs provided by the OSM community where different combinations of VIMs and NFVIs are running.

This permanent HIVE setup was connected to an additional instance of HIVE, HIVE-Nomad, which was deployed temporary at the 5TONIC Lab ahead of the remote integration and pre-testing phase.

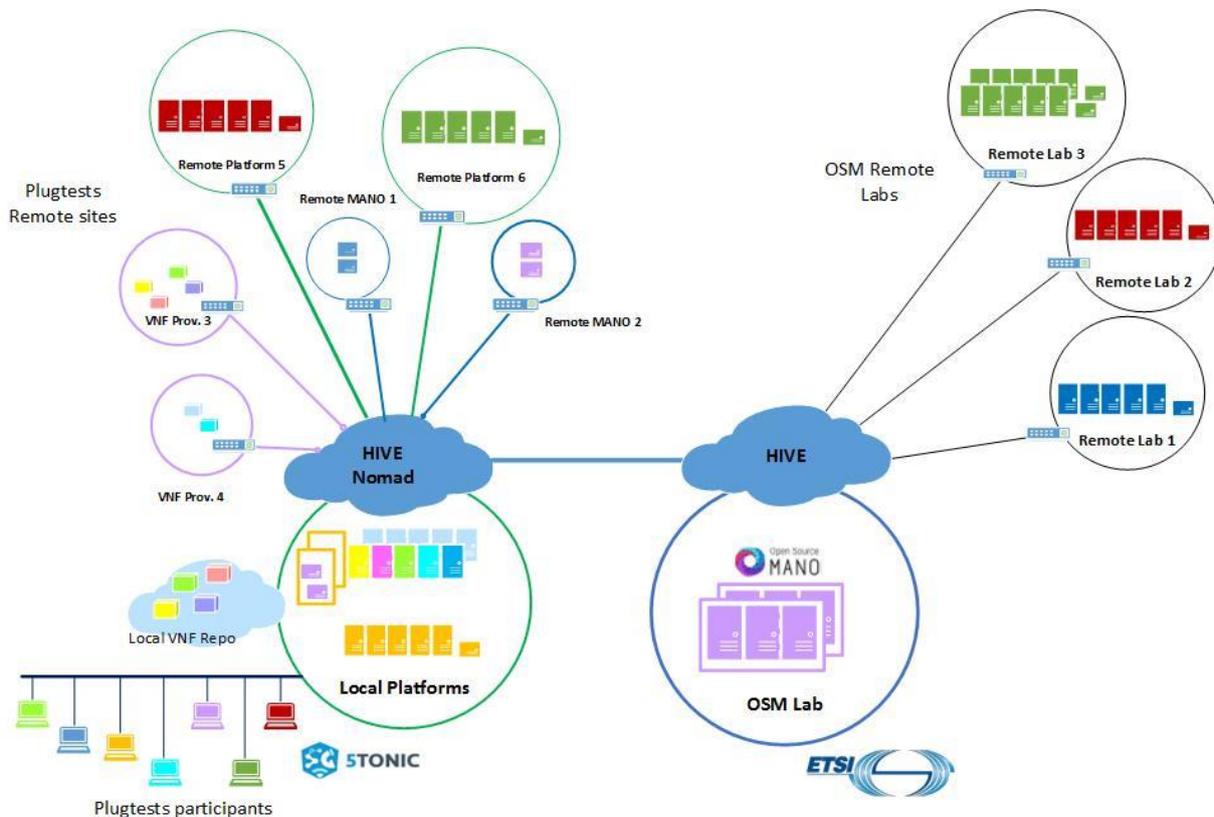
Coupled to the HIVE-nomad, a Network Attached Storage (NAS) was used as a central VNF repository to store all participating VNF and NS Packages, descriptors, Software Images, etc..

Once HIVE-nomad was deployed, a number of VPN tunnels were created to interconnect participants’ labs where the FUTs were running (MANOs and NFV Platforms). VNF providers also leveraged these VPN connection to HIVE-nomad to access the NAS and / or remote platforms for debugging purposes.

A total of 29 Remote Labs connected to the setup described above, either as a permanent OSM Remote Lab or as a Plugtests participant’s Lab.

## 6.2 Local Test Infrastructure

Once the remote integration and pre-testing phase completed, the Remote Test Infrastructure was extended to support local activities at the hosting lab, as follows:



**Figure 7. Local & Remote Test Infrastructure**

A local network was deployed at 5TONIC to allow participants to access the remote labs, the local NAS and some local instances of Hardware or Software that were deployed locally to support the testing. Two additional NFV Platforms (NFVI&VIM) were deployed locally in the 5TONIC lab and connected to the remote test infrastructure through HIVE.

In addition, 5TONIC provided a number of servers to host some test and support functions that were required by some VNFs or Network Services in order to successfully run the test plan.

---

## 7 Test Procedures

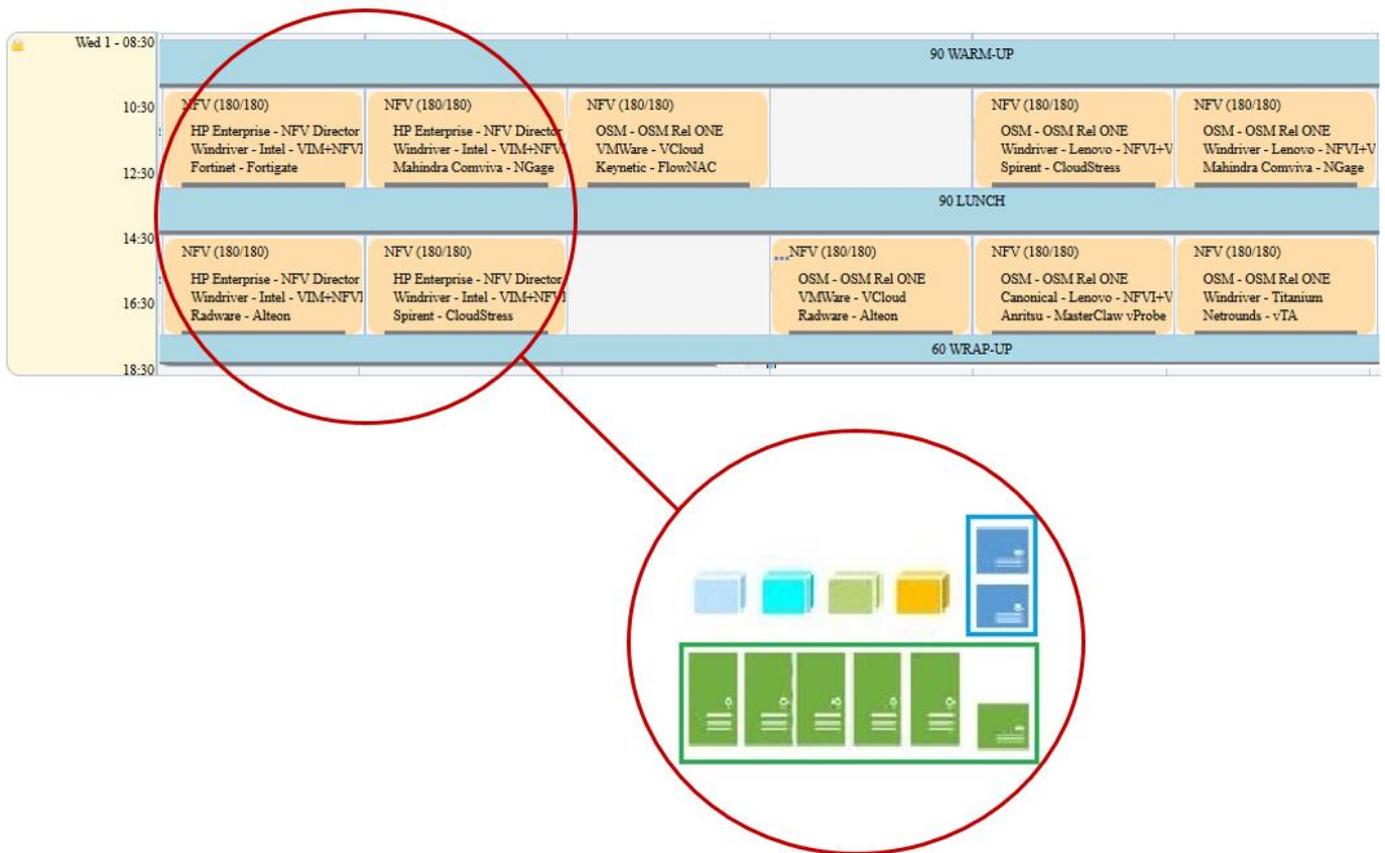
### 7.1 Remote Integration & Pre-testing Procedure

During the remote integration and pre-testing phase the following procedures were followed by the participation Functions Under Test. Once the FUT documentation and HIVE connection had been successfully completed, the following pre-testing activities were run:

- 1) MANO & VIM/NFVI:
  - a. Check connectivity and access rights
  - b. On-board and instantiate a reference VNF
  - c. Terminate reference VNF
- 2) VNF & MANO
  - a. Develop VNF and NS Descriptors
  - b. Upload Descriptors to the NAS (central repository)
  - c. On-board Descriptors to MANO
  - d. Instantiate NS/VNF on a reference VIM&NFVI
  - e. Terminate NFS/VNF
- 3) VNF & VIM
  - a. Upload SW images to the NAS (central repository)
  - b. On-board SW images in the VIM
  - c. Manually instantiate VNF
  - d. Terminate VNF

The progress of these procedures for the different combinations of FUTs (VNFs, MANOs, VIM&NFVIs) was captured in a 3 dimensional tracking matrix, and reviewed during the weekly calls. The tracking matrix was similar to the one shown below.





**Figure 9. Daily Schedule & Test Sessions**

The day started with a 1 hour and a half Warm-Up Session during which representatives of the different FUTs (MANO, NFVI&VIM and VNFs) that were scheduled together for the day went through a number of sanity checks in order to make sure the Test Sessions could be run smoothly. These checks included:

- 1) Verifying that the VNF Packages and NS Descriptors had been uploaded to the NAS.
- 2) Verifying that MANO had access to VNF and NS Descriptors in the NAS
- 3) Verifying MANO to VIM connectivity and accessibility (credentials, tenants, etc..)
- 4) Verifying that VNF Images had been uploaded to the VIM

During the Plugtests, additional verification tasks were added to the list, like verifying that the NFVI and VIM started from a clean state: i.e. VMs and processes from previous sessions and off-line testing had been properly stopped, removed and related resources released.

After the Warm-up session a 3 hour test session was run for on each MANO and VIM&NFVI for 1 or 2 VNFs each. An individual Test Session Report was recorded for each tested VNF.

After the morning Test Sessions and the lunch break, another series of 3 hour Test Sessions were run in the afternoon.

During each test session, for each tested VNF the Interoperability testing procedure was as follows:

- 1) VNF provider opened the Test Session Report and the Test Plan.

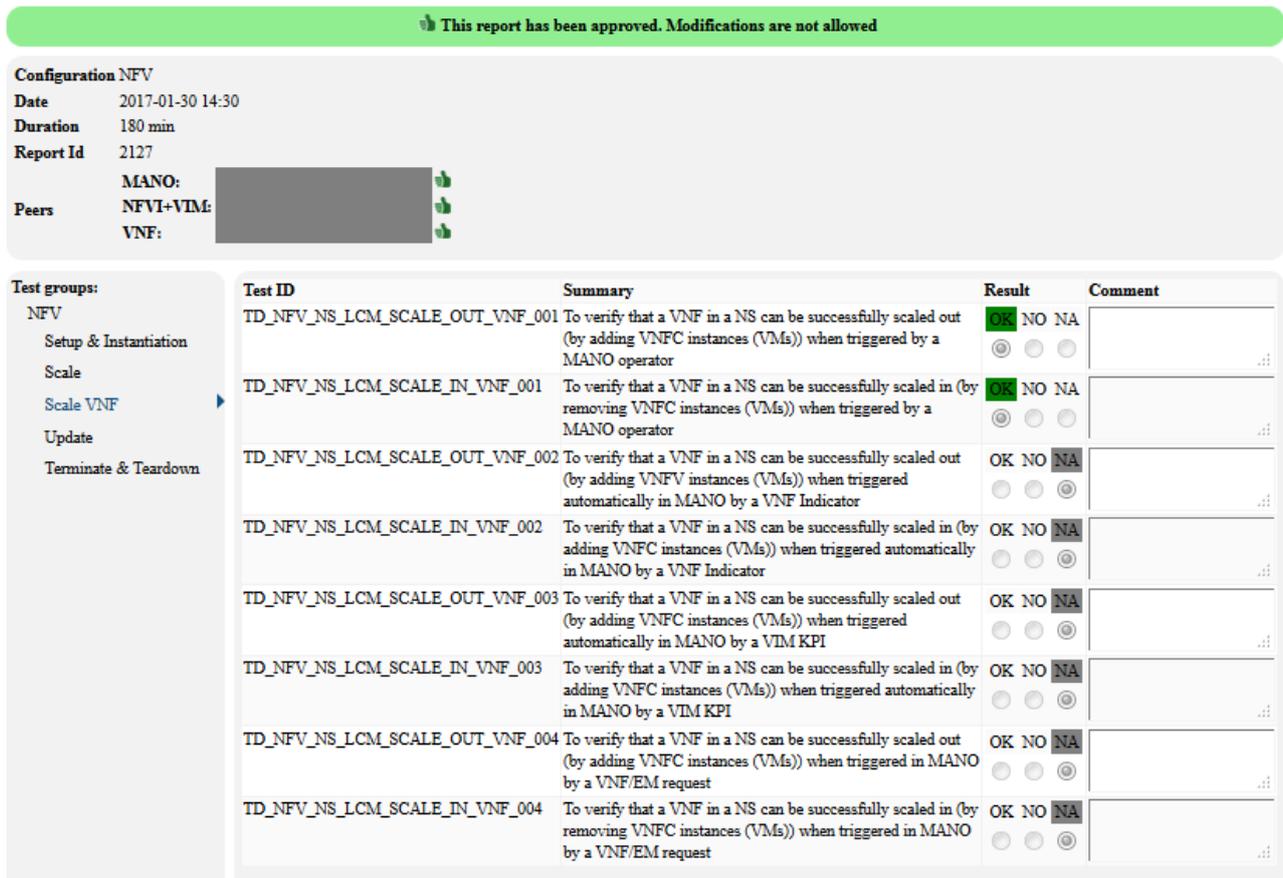


Figure 10. Test Session Report

2) For each Test in the Test Plan:

- a. The corresponding Test Description and SUT Configuration were followed.

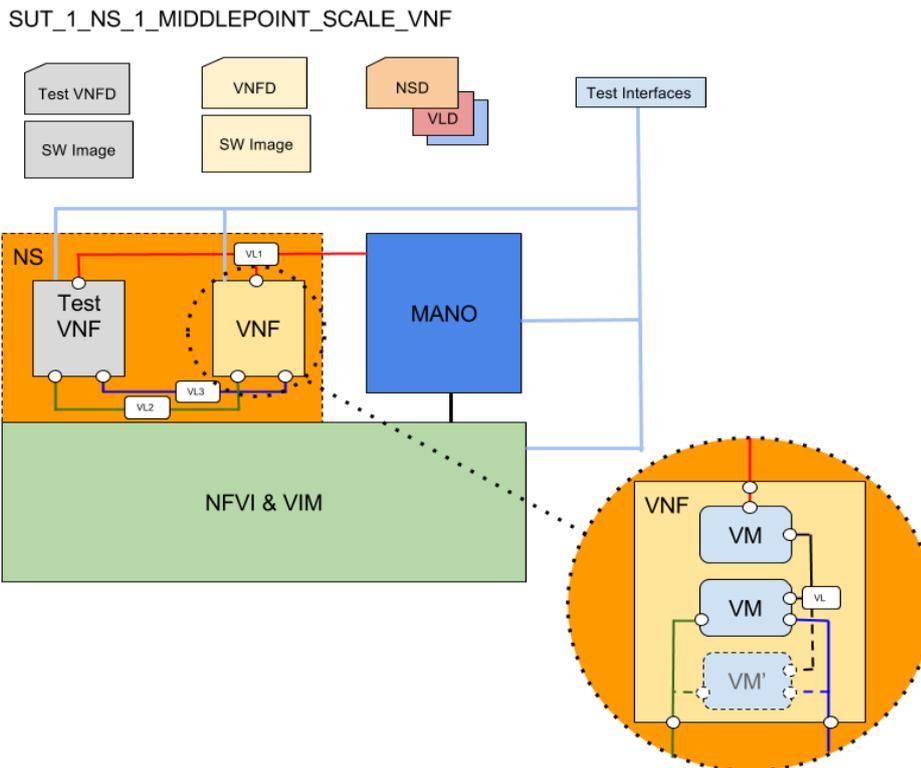


Figure 11. System Under Test (SUT) Configuration example.

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_OUT_VNF_001			
Test Purpose	To verify that a VNF in a NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered by a MANO operator			
Configuration	SUT_1_NS_1_ENDPOINT_SCALE_VNF SUT_1_NS_1_MIDDLEPOINT_SCALE_VNF			
References	ETSI GS NFV-IFA005 V2.1.1 (clause 5.3.4) ETSI GS NFV-IFA006 V2.1.1 (clauses 7.3.1, 7.4.1) ETSI GS NFV-IFA013 V2.1.1 (clause 7.3.4) ETSI GS NFV-IFA010 V2.1.1 (clauses 6.2.3, 6.3.3)			
Applicability	<ul style="list-style-type: none"> <li>* MANO can request VIM_NFVI to allocate virtualised resources</li> <li>* VIM_NFVI supports allocating virtualised resources</li> <li>* MANO supports triggering scale out with an operator's action</li> <li>* MANO supports scale out by adding VNFC instances (VMs)</li> <li>* NS/VNF supports scale out by adding VNF instances (VMs)</li> </ul>			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Trigger NS scale out (by adding VNFC instances (VMs)) to a VNF in the NS) in MANO with an operator action	
	2	IOP Check	Verify that the requested resources have been allocated by the VIM according to the descriptors	
	3	IOP Check	Verify that the additional VM(s) have been deployed (i.e by querying the VIM)	
	4	IOP Check	Verify that the additional VM(s) are running and are reachable through the management network	
	5	IOP Check	Verify that the additional VM(s) are connected to the VL(s) according to the descriptors	
	6	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test	
IOP Verdict				

Figure 12 Test Description example

- b. VNF, MANO and VIM&NFVI providers jointly executed the different steps specified in the test description and evaluated interoperability through the different IOP Checks prescribed in the Test Description
  - c. The VNF provider recorded the Test Result in the Test Session Report, as follows:
    - i.OK: all IOP Checks were successful
    - ii.NOK: at least one IOP Check failed. A comment was requested.
    - iii.NA: the feature was not supported by at least 1 of the involved FUTs. A comment was requested.
- 3) Once all the tests in the Test Session Report were executed and results recorded, the VNF, MANO and VIM&NFVI providers reviewed the Report and approved it.

## 8 Test Plan Overview

### 8.1 Introduction

This 1<sup>st</sup> NFV Plugtests Test Plan was developed by ETSI Centre for Testing and Interoperability following the interoperability testing methodology defined by ETSI NFV in [TST002].

The Test Plan was reviewed and discussed with participants during the Plugtests preparation and pre-testing phase. The Test Descriptions were organised in groups covering the NFV capabilities in scope, extracted from NFV Release 2.

The complete Test Plan has been submitted to ETSI NFV TST Working Group and can be found in the ETSI Portal, see [1NFVPLU-TP] in Clause 2.

The following clauses summarise the 26 test cases in scope for this Plugtests, organised by Test Groups

### 8.2 Test Groups

#### 8.2.1 Setup

This Group targets the on boarding by MANO on the VIM&NFVI of the VNF Package(s) and Network Service Descriptor allowing to validate proper behaviour of the VNF(s) Under Test.

Test Id	Test Purpose
TD_NFV_SETUP_ONBOARD_VNF_PKG_001	To on-board a VNF Package
TD_NFV_SETUP_ONBOARD_NSD_001	To on-board a NSD

**Table 6. Test Group: Setup**

#### 8.2.2 Network Service Life Cycle Management

##### 8.2.2.1 Instantiate

This Group targets the instantiation by MANO on the VIM&NFVI of the previously on boarded Network Service including the VNF(s) Under Test and Test VNF(s).

Test Id	Test Purpose
TD_NFV_NS_LCM_INSTANTIATE_001	To verify that an NS can be successfully instantiated

**Table 7. Test Group: NS LCM - Instantiate**

An example of System Under Test Configuration in scope for this Group is provided here after:

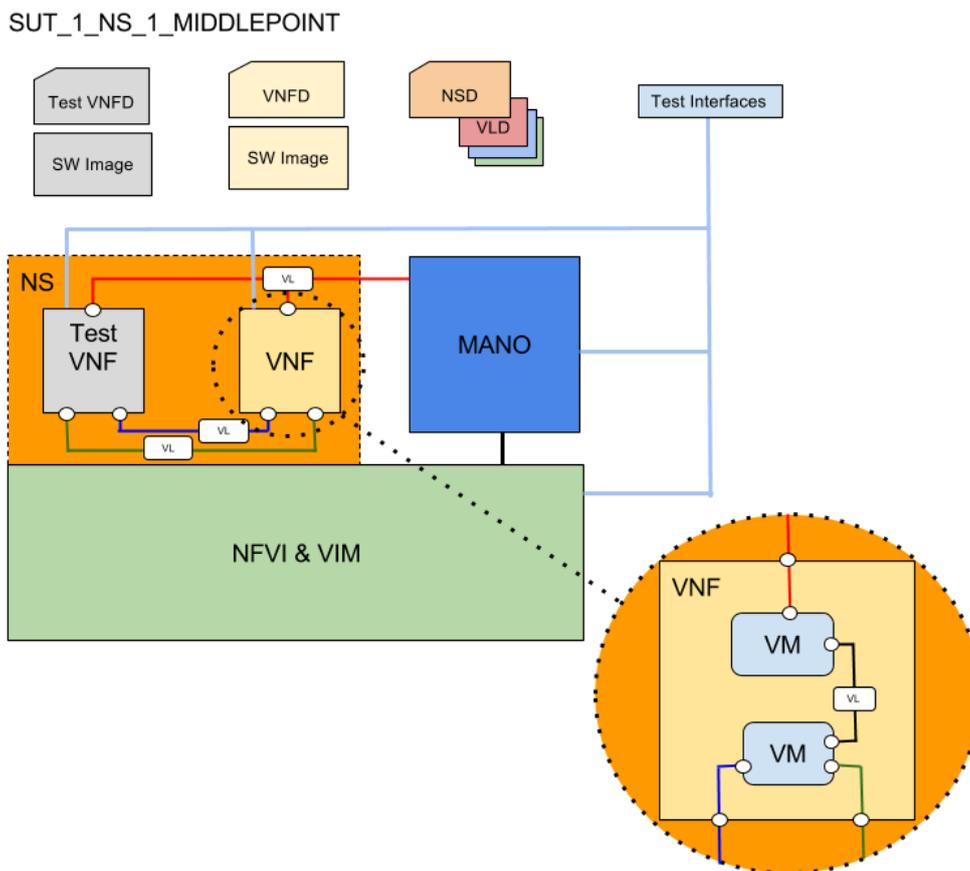


Figure 13 SUT Configuration example for a middle point VNF

### 8.2.2.2 Scale

This Group targets Scaling Out and In operations by adding VNF Instances to the Network Service. As detailed in the following table, 4 different triggers were foreseen for these scaling operations: manual operation in MANO, VNF Indicator, VIM KPI and explicit request from VNF/EM.

Test Id	Test Purpose
TD_NFV_NS_LCM_SCALE_OUT_001	To verify that a NS can be successfully scaled out (by adding VNF instances) if triggered by a MANO operator
TD_NFV_NS_LCM_SCALE_IN_001	To verify that a NS can be successfully scaled in (by removing VNF instances) if triggered by a MANO operator
TD_NFV_NS_LCM_SCALE_OUT_002	To verify that a NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VNF Indicator
TD_NFV_NS_LCM_SCALE_IN_002	To verify that a NS can be successfully scaled in (by removing VNF instances) if triggered automatically in MANO by a VNF Indicator
TD_NFV_NS_LCM_SCALE_OUT_003	To verify that a NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VIM KPI
TD_NFV_NS_LCM_SCALE_IN_003	To verify that a NS can be successfully scaled in (by removing VNF instances) if triggered automatically in MANO by a VIM KPI
TD_NFV_NS_LCM_SCALE_OUT_004	To verify that a NS can be successfully scaled out (by adding VNF instances) if triggered in MANO by a VNF/EM request
TD_NFV_NS_LCM_SCALE_IN_004	To verify that a NS can successfully scale in (by removing VNF instances) if triggered in MANO by a VNF/EM request

Table 8. Test Group: NS LCM - Scale

An example of System Under Test Configuration in scope for this Group is provided here after:

SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE

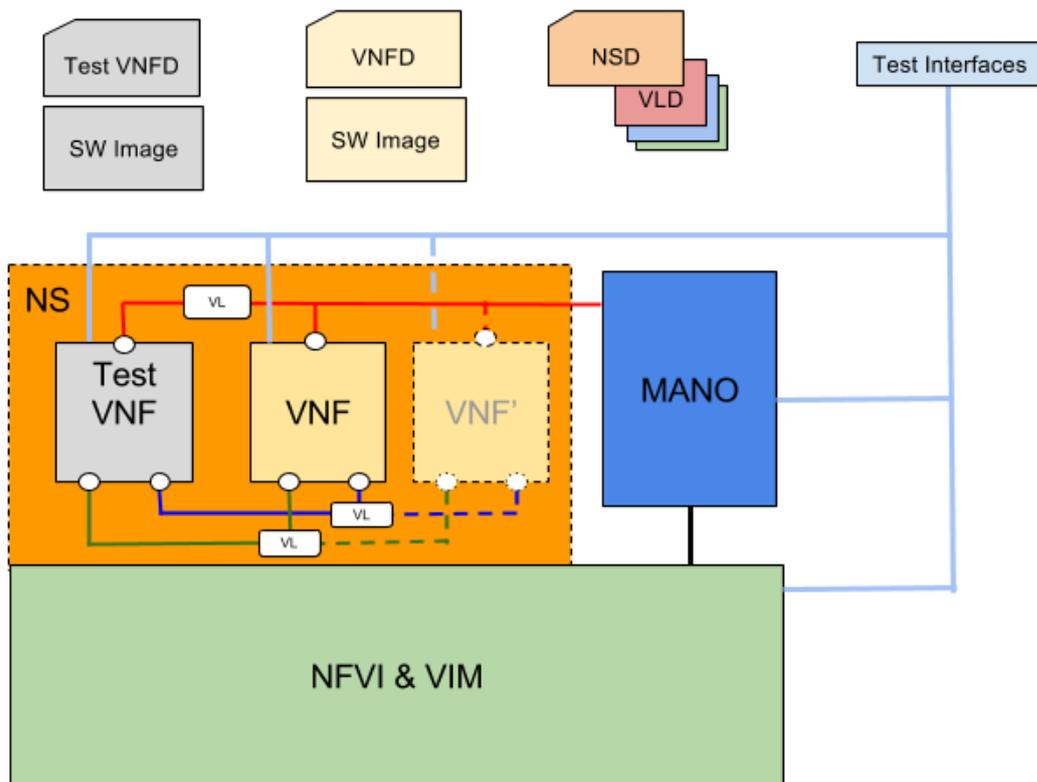


Figure 14. SUT Configuration example for scaling a middle point VNF (+/- VNFs)

8.2.2.3 Scale VNF

This Group targets Scaling Out and In operations by adding VNFC Instance(s) to the VNF(s) in the Network Service.

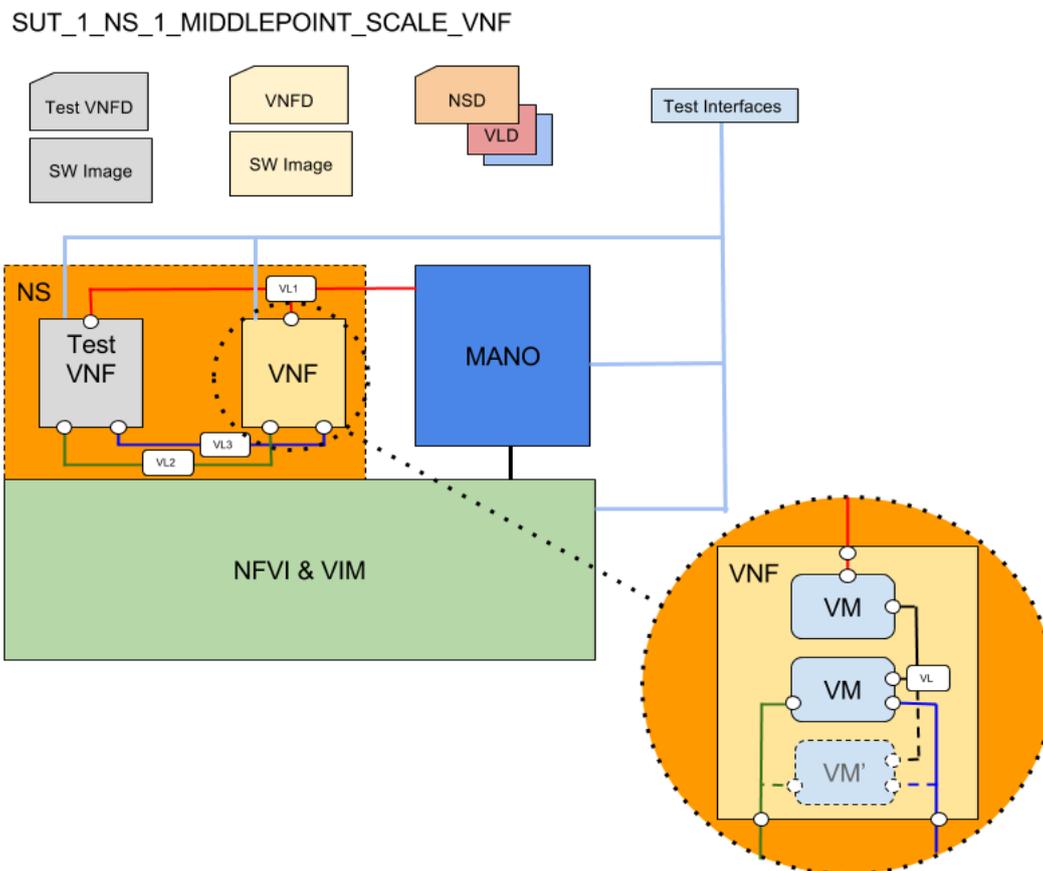
As detailed in the following table, the same 4 different triggers described in the previous clause were foreseen for these scaling operations: manual operation in MANO, VNF Indicator, VIM KPI and explicit request from VNF/EM.

Test Id	Test Purpose
TD_NFV_NS_LCM_SCALE_OUT_VNF_001	To verify that a VNF in a NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered by a MANO operator
TD_NFV_NS_LCM_SCALE_IN_VNF_001	To verify that a VNF in a NS can be successfully scaled in (by removing VNFC instances (VMs)) when triggered by a MANO operator
TD_NFV_NS_LCM_SCALE_OUT_VNF_002	To verify that a VNF in a NS can be successfully scaled out (by adding VNFV instances (VMs)) when triggered automatically in MANO by a VNF Indicator
TD_NFV_NS_LCM_SCALE_IN_VNF_002	To verify that a VNF in a NS can be successfully scaled in (by adding VNFC instances (VMs)) when triggered automatically in MANO by a VNF Indicator
TD_NFV_NS_LCM_SCALE_OUT_VNF_003	To verify that a VNF in a NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered automatically in MANO by a VIM KPI
TD_NFV_NS_LCM_SCALE_IN_VNF_003	To verify that a VNF in a NS can be successfully scaled in (by adding VNFC instances (VMs)) when triggered automatically in MANO by a VIM KPI
TD_NFV_NS_LCM_SCALE_OUT_VNF_004	To verify that a VNF in a NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered in MANO by a VNF/EM request

TD_NFV_NS_LCM_SCALE_IN_VNF_004	To verify that a VNF in a NS can be successfully scaled in (by removing VNFC instances (VMs)) when triggered in MANO by a VNF/EM request
--------------------------------	--

**Table 9. Test Group: NS LCM – Scale VNF**

An example of System Under Test Configuration in scope for this Group is provided here after:



**Figure 15. SUT Configuration example for scaling a middle point VNF (+/- VNFCs)**

### 8.2.2.4 Update

This Group targets 2 different types of Network Service Update operations:

- 1) VNF Stop/Re-start: The SUT Configurations are the same used for the Instantiate operation.
- 2) Update NS by adding/removing VNFs and VLs. An example of SUT configuration is provided hereafter:

Test Id	Test Purpose
TD_NFV_NS_LCM_UPDATE_STOP_VNF_001	To verify that a VNF running in a NS can be successfully stopped by MANO
TD_NFV_NS_LCM_UPDATE_START_VNF_001	To verify that a stopped VNF in a NS can be successfully re-started by MANO
TD_NFV_NS_LCM_UPDATE_ADD_VNF_VL_001	To verify that VNF(s) and VL(s) can be added to a running NS
TD_NFV_NS_LCM_UPDATE_REM_VNF_VL_001	To verify that VNF(s) and VL(s) can be removed from a running NS

**Table 9. Test Group: NS LCM - Update**

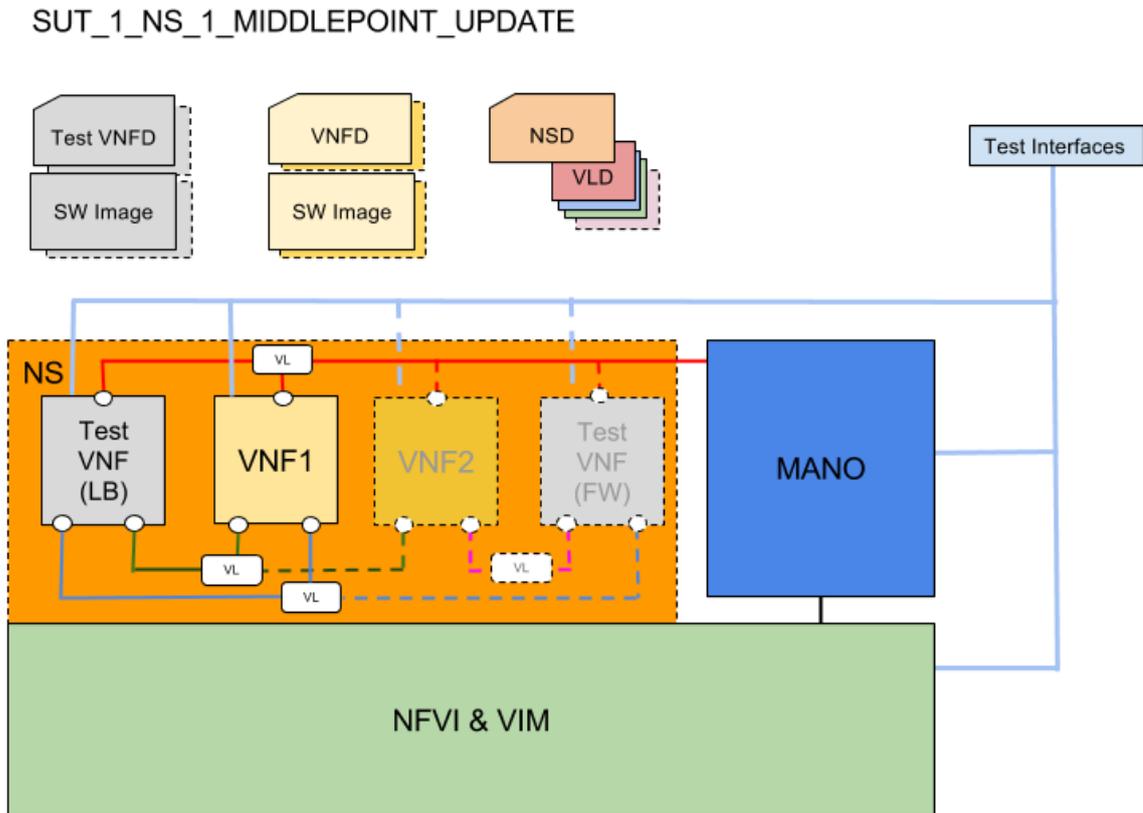


Figure 16. SUT Configuration example for Update NS (+/- VNF&VL)

### 8.2.2.5 Terminate

This Group targets termination of the Network Service. The SUT Configurations are the same used for the Instantiate operation.

Test Id	Test Purpose
TD_NFV_NS_LCM_TERMINATE_001	To verify that a NS can be successfully terminated

Table 10. Test Group: NS LCM - Terminate

### 8.2.3 Teardown

This Group targets deletion of the NSD and VNF Package.

Test Id	Test Purpose
TD_NFV_TEARDOWN_DELETE_NS_001	To delete a NSD
TD_NFV_TEARDOWN_DELETE_VNF_PKG_001	To delete a VNF Package

Table 11. Test Group: Tear Down

## 9 Interoperability Results

### 9.1 Overall Results

During the Plugtests, a total of 160 Test Sessions were run: that is, 160 different combinations of the 3 Functions Under Test in scope: VNF, MANO and VIM&NFVI were tested for interoperability on a subset of NFV Release 2 capabilities. Overall, a total of 1554 individual test cases were run and reported interoperability results.

The table below provides the overall results (aggregated data) from all the test cases run during all the Test Sessions with all the different combinations of Functions Under Test from all the participating companies.

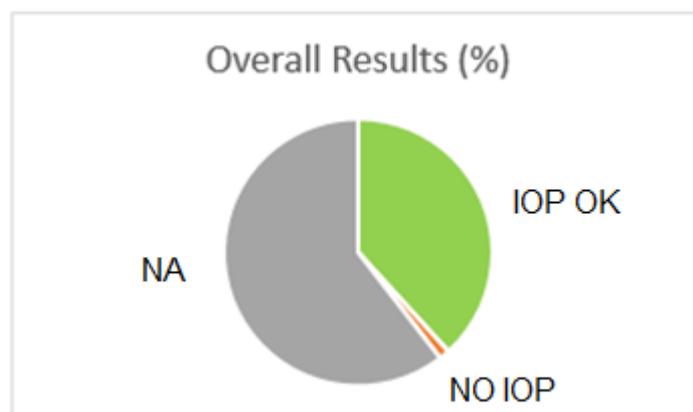
During each Test Session a maximum of 26 individual results could be reported, corresponding to the 26 Test Cases in the Test Plan. Through the 160 Test Sessions run, a total of 3985 Test Results were reported from the possible 4160 (26 \* 160). This figure includes both the executed and non-executed Test Cases.

Among the executed Test Cases, the possible results were “OK”, when interoperability was successfully achieved and “NO” (Not OK) when it was not. The non-executed Test Cases were marked “NA” (Not Applicable) during the Test Session, to indicate that at least one of the FUTs involved in the Test Session did not support the feature in scope. Finally, the tests cases for which no result was reported (i.e test session run out of time) were not taken into account in the Total Results figure. The Total Results represents the total amount of test cases for which results were reported.

Interoperability		Not Executed	Totals	
OK	NO	NA	Run	Results
1539 (99.0%)	15 (1.0%)	2431 (61.0%)	1554 (39.0%)	3985

**Table 12: Overall Results**

It is worth noting that the Test Plan was designed to support a series of NFV Plugtests and included not only Network Service on-boarding and deployment test cases but also more complex scenarios targeting Network Service updates and scaling. Despite this very ambitious Test Plan, the overall execution rate in this first Plugtests was close to 40%.



**Figure 17. Overall results (%)**

Among the executed test cases, the overall interoperability rate is 99%, which indicates a very high degree of compatibility among the participating implementations (FUTs) in the areas of the Test Plan where features were widely supported and the test cases could be executed in most of the Test Sessions. In the next clauses, we will see that this high rate is also a consequence of the good preparation and involvement of participants during the remote integration and pre-testing phase of the Plugtests.

The next clauses present more detailed results per test group and test cases and will allow to identify the areas and features with higher execution and interoperability rates.

## 9.2 Results per Test Group

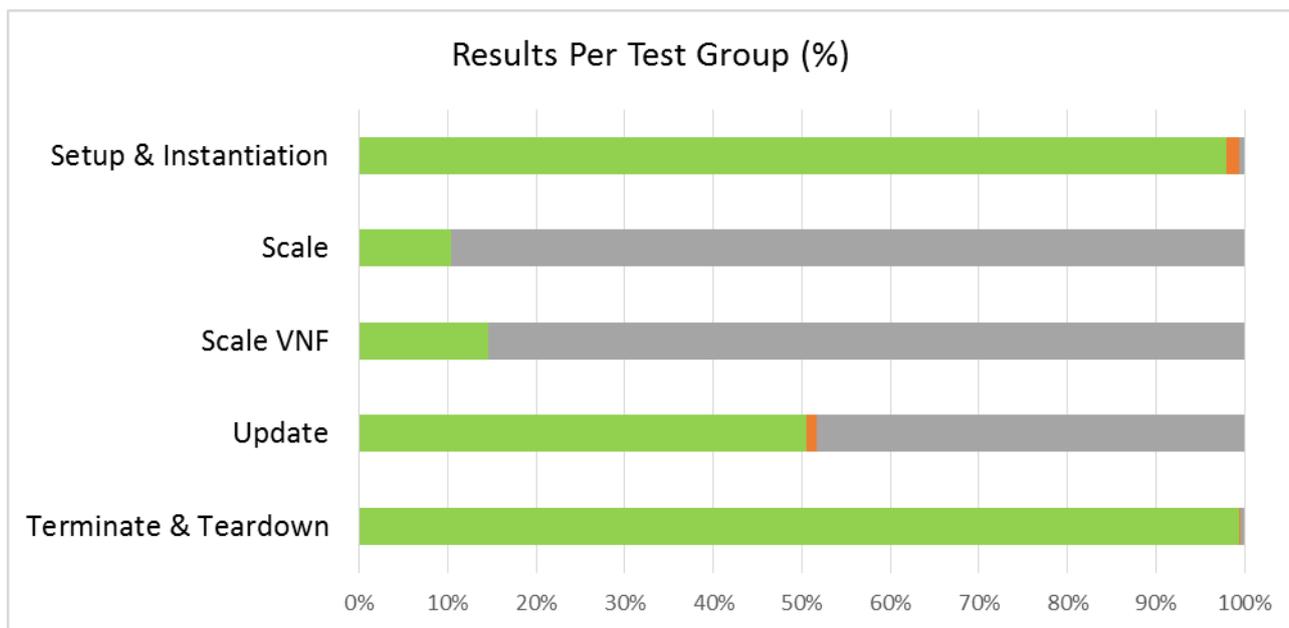
The table below provides the results for each test group in the scope of the Plugtests.

Test Group	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
Setup & Instantiation	468 (98.5%)	7 (1.5%)	3 (0.6%)	475 (99.4%)	478
Scale	126 (100.0%)	0 (0.0%)	1096 (89.7%)	126 (10.3%)	1222
Scale VNF	178 (100.0%)	0 (0.0%)	1038 (85.4%)	178 (14.6%)	1216
Update	305 (97.7%)	7 (2.3%)	292 (48.3%)	312 (51.7%)	604
Terminate & Teardown	462 (99.8%)	1 (0.2%)	2 (0.4%)	463 (99.6%)	165

**Table 13. Results per Test Group**

The table shows very high execution and interoperability rates (over 98%) for the Setup, Instantiation, Terminate and Teardown Test Groups. These features were part of the main scope of the pre-testing sessions, which in the context of new technologies, and in the absence of fully adopted standards, are key to achieve interoperability.

With regards to the more complex features, like the ones targeted by the Scale, Scale VNF and Update Test Groups, a more detailed look at the results per Test Case is required to derive conclusions.



**Figure 18. Results per Test Group (%)**

## 9.3 Results per Test Case

The Figure below provides an overview of the execution, success and failure rates for each individual test case:

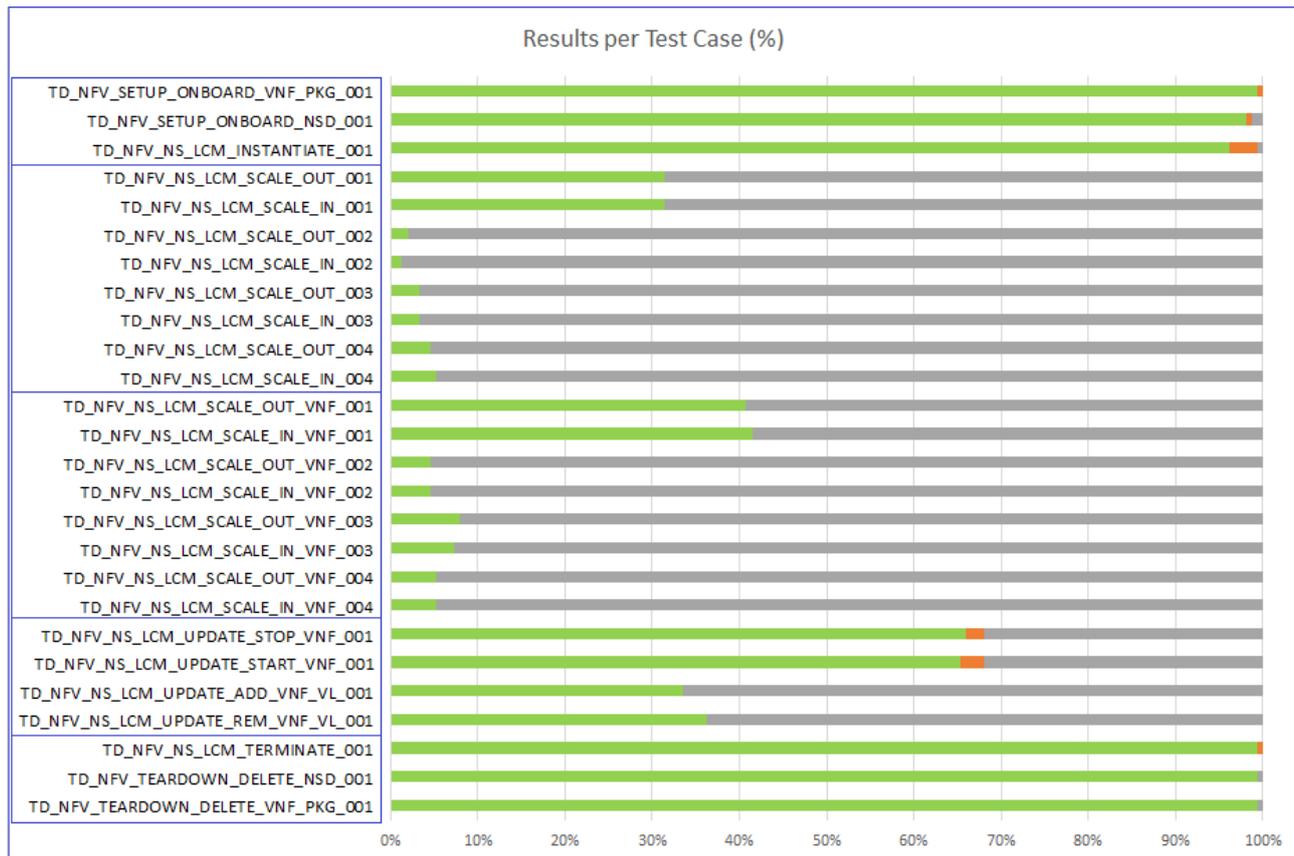


Figure 19. Results per Test Case (%)

The next clauses, tables and diagrams provide the exact execution and interoperability rates for each individual test case, organized by test group. When possible, some conclusions have been derived.

### 9.3.1 Test Group: Setup & Instantiation

The results for the Test Cases in this group are as follows.

Test Group: Setup & Instantiation	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_NFV_SETUP_ONBOARD_VNF_PKG_001	159 (99.4%)	1 (0.6%)	0 (0.0%)	160 (100.0%)	160
TD_NFV_SETUP_ONBOARD_NSD_001	156 (99.4%)	1 (0.6%)	2 (1.3%)	157 (98.7%)	159
TD_NFV_NS_LCM_INSTANTIATE_001	153 (96.8%)	5 (3.2%)	1 (0.6%)	158 (99.4%)	159

Table 14. Results per Test Case (Setup & Instantiation)

The execution rate of all the test cases in this group is above 98 % which indicates a consistent support of these operations by all the participating FUTs.

The interoperability rates for VNF Package and NS on-boarding operations are over 99 % through all the combinations of FUTs that were tested, indicating a successful integration and high degree of interoperability among all the participating FUTs for these operations.

The interoperability rate for the Network Service Instantiation was slightly lower, but still a very satisfactory 96.8%. An analysis of the failures indicates that most of them came from networking requirements from the VNF not being implemented in the VIM&NFVI, or the MANO and VIM implementing incompatible solutions for such requirements.

### 9.3.2 Test Group: Scale

This Test Group included Scale Out and In operations by adding VNF Instances to the Network Service. As described in the test plan, 4 different triggers were foreseen for these scaling operations:

- 001 – Scale operation triggered manually in MANO by an operator
- 002 – Scale operation triggered automatically in MANO by the reception of a VNF Indicator from the VNF/EM
- 003 – Scale operation triggered automatically in MANO by the reception of a KPI from the VIM
- 004 – Scale operation triggered in MANO by the reception of a request from VNF/EM.

The following table summarizes the results for the Test Cases in this group.

Test Group: Scale OUT/IN (by adding VNFs)	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_NFV_NS_LCM_SCALE_OUT_001	48 (100.0%)	0 (0.0%)	105 (68.6%)	48 (31.4%)	153
TD_NFV_NS_LCM_SCALE_IN_001	48 (100.0%)	0 (0.0%)	105 (68.6%)	48 (31.4%)	153
TD_NFV_NS_LCM_SCALE_OUT_002	3 (100.0%)	0 (0.0%)	150 (98.0%)	3 (2.0%)	153
TD_NFV_NS_LCM_SCALE_IN_002	2 (100.0%)	0 (0.0%)	151 (98.7%)	2 (1.3%)	153
TD_NFV_NS_LCM_SCALE_OUT_003	5 (100.0%)	0 (0.0%)	148 (96.7%)	5 (3.3%)	153
TD_NFV_NS_LCM_SCALE_IN_003	5 (100.0%)	0 (0.0%)	147 (96.7%)	5 (3.3%)	152
TD_NFV_NS_LCM_SCALE_OUT_004	7 (100.0%)	0 (0.0%)	145 (95.4%)	7 (4.6%)	152
TD_NFV_NS_LCM_SCALE_IN_004	8 (100.0%)	0 (0.0%)	-145 (94.8%)	8 (5.2%)	153

**Table 15. Results per Test Case (Scale)**

This group, which covered one of the complex features of the Test Plan, required consistent support of Scaling operation by the 3 FUTs involved in each Test Session. Despite Scaling not being explicitly in scope of the pre-testing phase, manually triggered scaling could be successfully tested in over 30% of the FUT combinations. Moreover, in those combinations where the scaling could be tested, full interoperability (100%) was reported.

Also, Plugtests participants reported that for some complex VNFs including multiple VNF Components, scaling the whole VNF didn't really make sense, so they only targeted scale operations by adding / removing VNF Components (next Group), and reported test cases in this group as not applicable (NA).

### 9.3.3 Test Group: Scale VNF

This Test Group included Scaling Out and In operations by adding VNFV Instances to the VNF. As in the previous Group, 4 different triggers were foreseen for these scaling operations:

- 001 – Scale operation triggered manually in MANO by an operator
- 002 – Scale operation triggered automatically in MANO by the reception of a VNF Indicator from the VNF/EM
- 003 – Scale operation triggered automatically in MANO by the reception of a KPI from the VIM
- 004 – Scale operation triggered in MANO by the reception of a request from VNF/EM.

The following table summarizes the results for the Test Cases in this group.

Test Group: Scale OUT/IN (by adding VNFs)	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_NFV_NS_LCM_SCALE_OUT_VNF_001	62 (100.0%)	0 (0.0%)	90 (59.2%)	62 (40.8%)	152
TD_NFV_NS_LCM_SCALE_IN_VNF_001	63 (100.0%)	0 (0.0%)	89 (58.6%)	63 (41.4%)	152
TD_NFV_NS_LCM_SCALE_OUT_VNF_002	7 (100.0%)	0 (0.0%)	145 (95.4%)	7 (4.6%)	152
TD_NFV_NS_LCM_SCALE_IN_VNF_002	7 (100.0%)	0 (0.0%)	145 (95.4%)	7 (4.6%)	152
TD_NFV_NS_LCM_SCALE_OUT_VNF_003	12 (100.0%)	0 (0.0%)	140 (92.1%)	12 (7.9%)	152
TD_NFV_NS_LCM_SCALE_IN_VNF_003	11 (100.0%)	0 (0.0%)	141 (92.8%)	11 (7.2%)	152
TD_NFV_NS_LCM_SCALE_OUT_VNF_004	8 (100.0%)	0 (0.0%)	144 (94.7%)	8 (5.3%)	152
TD_NFV_NS_LCM_SCALE_IN_VNF_004	8 (100.0%)	0 (0.0%)	144 (94.7%)	8 (5.3%)	152

**Table 16. Results per Test Case (Scale VNF)**

For this group, which also targets a complex feature that was not explicitly in scope of the pre-testing phase, the aggregated data shows an increase of the execution rate to 40% of the FUT combinations for manually triggered scaling. Again, in those combinations where this type of scaling could be tested, full interoperability (100%) was reported.

### 9.3.4 Test Group: Update

This Test Group covered 2 distinct types of Network Service Update operations:

- 1) Stop/Re-start VNF
- 2) Update NS by adding/removing VNF(s) and VL(s)

Test Group: NS Update	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_NFV_NS_LCM_UPDATE_STOP_VNF_001	101 (97.1%)	3 (2.9%)	49 (32.0%)	104 (68.0%)	153
TD_NFV_NS_LCM_UPDATE_START_VNF_001	100 (96.1%)	4 (3.8%)	49 (32.0%)	104 (68.0%)	153
TD_NFV_NS_LCM_UPDATE_ADD_VNF_VL_001	50 (100.0%)	0 (0.0%)	99 (66.4%)	50 (33.6%)	149
TD_NFV_NS_LCM_UPDATE_REM_VNF_VL_001	54 (100.0%)	0 (0.0%)	95 (63.8%)	54 (36.2%)	149

**Table 17. Results per Test Case (NS Update)**

The execution rate for the first type of Network Service Update was close to 70% despite these Test Cases being added relatively late to the Test Plan and not being addressed during the pre-testing phase. The Interoperability rate was above 96%. The reported failures were related to major differences in VNF and MANO implementations of the VNF stop/re-start operations. The different possible interpretations of these operations were discussed during the Plugtests and are summarized in Clause 10 Plugtests Outcome.

The execution rate for the second type of Network Service Update was around 34-36%. Once more these Test Cases arrived late to the Test Plan and were not enforced during the pre-testing phase. That being said, in the FUT combinations where these Test Cases could be run, full interoperability (100%) was reported.

### 9.3.5 Test Group: Terminate & Teardown

This last Test Group covers Network Service Termination and descriptors removal.

Test Group: Terminate & Teardown	Interoperability		Not Executed	Totals	
	OK	NO	NA	Run	Results
TD_NFV_NS_LCM_TERMINATE_001	154 (99.4%)	1 (0.6%)	0 (0.0%)	155 (100.0%)	155
TD_NFV_TEARDOWN_DELETE_NSD_001	154 (100.0%)	0 (0.0%)	1 (0.6%)	154 (99.4%)	155
TD_NFV_TEARDOWN_DELETE_VNF_PKG_001	154 (100.0%)	0 (0.0%)	1 (0.6%)	154 (99.4%)	155

**Table 18. Results per Test Case (Terminate & Teardown)**

The execution and interoperability rates for all the Test Cases in this group are above 99% which indicates a consistent support of these operations by all the participating FUTs.

## 10 Plugtests Outcome

### 10.1 Feedback on the Test Plan

#### 10.1.1 NS Update – Stop / Re-start VNF

The Network Service Update – Stop/Re-start VNF Test Cases (TD\_NFV\_NS\_LCM\_UPDATE\_STOP\_VNF\_001 and TD\_NFV\_NS\_LCM\_UPDATE\_START\_VNF\_001) generated some discussion among the Plugtests participants.

Interoperability Test Description				
<b>Identifier</b>	TD_NFV_NS_LCM_UPDATE_STOP_VNF_001			
<b>Test Purpose</b>	To verify that a VNF running in a NS can be successfully stopped by MANO			
<b>Configuration</b>	SUT_1_NS_1_ENDPOINT SUT_1_NS_1_MIDDLEPOINT			
<b>References</b>	ETSI GS NFVIFA013 V2.1.1 (clause 7.3.5)			
<b>Applicability</b>	* MANO can request VM_NFM to stop VM(s) * VM_NFM supports stopping running VM(s)			
<b>Pre-test conditions</b>	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * VNF instance(s) in the NS are running			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the VNF(s) stop operation in MANO	
	2	IOP Check	Verify the VNF(s) state inside the NS is "Stopped" on MANO (query, display, ...)	
	3	IOP Check	Verify that individual VM(s) inside the VNF(s) are stopped on VM (i.e query or display the state from VM)	
<b>IOP Verdict</b>				

**Figure 20. TD\_NFV\_NS\_LCM\_UPDATE\_STOP\_VNF\_001**

While the last step in the Test Description asks to verify that the VNFC instances (VMs) had been stopped, some participants had interpreted the stop/re-start operations at the VNF service level. A detailed analysis of the corresponding NFV specifications (NFV-IFA0013) did not allow to clarify the exact intention of the operation and get to a conclusion.

Moreover, the analysis triggered additional discussion on the applicability of the VNF Instance state transitions diagram provided in NFV-SWA001: *“The stop and start operation should be seen from the perspective of the VNF state diagram. Here the VNF can only be started when it reached the configured-inactive state. If a VNF is in a not-configured state and gets its configuration from the EM, it must have the capability to exchange messages with EM. If the inactive state would be considered as disabling the CPU resources, the EM cannot configure the VNF. So the VNF must be in a state where the vCPU resources are still ON but the VNF does not receive or process any packets. Packets should be received and processed only when the VNF is in the state "active". The state transition caused by the command start and stop is not to control the vCPU resources. It is the control if the VMP processes packets or not. This can be done by two possible solutions: (a) the VNF controls whether to process packets or not or (b) the NFI controls whether the VNF can process packets”*

For the scope of the 1<sup>st</sup> NFV Plugtests it was agreed to keep the TDs as they were, and interpret “stop VNF” as “shutdown” of this components (VMs). It should be noted that “shutdown” would be the term most commonly used in cloud environments. It was also agreed that these TDs could be extended to cover the graceful shutdown case which would include additional interaction between MANO and the VNF.

It was also decided to report the ambiguities to ETSI NFV, see clause 10.2 here after.

ETSI NFV’s clarification on the scope of the stop/re-start VNF and the applicable VNF Instance state transitions diagram could have an impact on these Test Descriptions and should be taken into account for the next Plugtests by

adapting them to the actual scope of the Stop/Re-start VNF operations or by adding new Test Description to cover Stop/Re-start VNF at a service level (if applicable).

### 10.1.2 SFC Testing

During the Plugtests there were several discussions on SFC Testing. While initially some specific TDs had been developed to cover one specific case of Service Function Chaining implementation, after discussion it was agreed that the existing TD for NS Instantiation could already be used as high level guidance for testing any kind of Network Service. The table below suggests some extensions (in *blue*) to the existing TD\_NFV\_NS\_LCM\_INSTANTIATE\_001 that could help to provide some additional guidance for some concrete cases of SFC testing

Interoperability Test Description				
<b>Identifier</b>	TD_NFV_NS_LCM_INSTANTIATE_0042			
<b>Test Purpose</b>	To verify that an NS <i>with NSH based SFC</i> can be successfully instantiated			
<b>Configuration</b>	<i>SUT_1_NS_1_ENDPOINT</i> SUT_1_NS_1_MIDDLEPOINT			
<b>References</b>	ETSI GS NFV-IFA013 V2.1.1 (clause 7.3.3) ETSI GS NFV-IFA005 V2.1.1 (clause 7.2.4) ETSI GS NFV-IFA006 V2.1.1 (clause 7.2.3) ETSI GS NFV-IFA008 V2.1.1 (clause 6.2.2) ETSI GS NFV-IFA010 V2.1.1 (clause 6.3.2) <i>ietf RFC 7665 SFC <a href="https://datatracker.ietf.org/doc/rfc7665/">https://datatracker.ietf.org/doc/rfc7665/</a></i> <i>ietf NSH draft <a href="https://datatracker.ietf.org/doc/draft-ietf-sfc-nsh/">https://datatracker.ietf.org/doc/draft-ietf-sfc-nsh/</a></i>			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• MANO can request VIM_NFVI to add a SW image</li> <li>• VIM_NFVI supports adding a SW image</li> <li>• MANO can request VIM_NFVI to allocate virtualised resources</li> <li>• VIM_NFVI supports allocating virtualised resources</li> <li>• (If required by NSD) MANO can request VIM_NFVI to create NFP(s)</li> <li>• (If required by NSD) VIM_NFVI supports creating NFP(s)</li> <li>• <i>NFVI_VIM supports NSH</i></li> <li>• <i>VNF supports Network Service Headers (NSH) encapsulation</i></li> <li>• <i>MANO supports SFC</i></li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NSD, VLD(s), VNFFGD(s) and VNF Package(s) have been on-boarded in MANO</li> <li>• The software image repository is reachable by the VIM</li> <li>• The required resources are available on the NFVI</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	stimulus	Trigger NS instantiation in MANO	
	2	IOP check	Verify that the software images have been on-boarded in the VIM	
	3	IOP check	Verify that the requested resources have been allocated by the VIM according to the descriptors	
	4	IOP check	Verify that the VNF (s) have been deployed according to the descriptors (VMs, VLs, CPs...)	
	5	IOP check	Verify that the VL and VNFFG instance(s) have been created according to the descriptors	
	6	IOP check	Verify that the VNF(s) are running and reachable through the management network	
	7	IOP check	Verify that the VNF(s) have been configured according to VNFD(s) (i.e by obtaining a result from the management interface)	
	8	IOP check	Verify that the VNF(s), VL(s) and VNFFG(s) have been connected according to the Descriptors	
	9	IOP check	Verify that the NS is successfully instantiated by running the end-to-end functional test ( <i>NSH Traffic</i> )	
<b>IOP Verdict</b>				

Figure 22. TD\_NFV\_NS\_LCM\_INSTANTIATE\_001 extension proposal

Similar extensions of basic TDs could be considered in the future to cover other specific cases of Network Services, i.e. EPA (Enhanced Platform Awareness) support, etc...

## 10.2 Feedback on NFV Specifications

### 10.2.1 NFV-IFA013. Update NS (Operate VNF)

The Network Service Update – Stop/Re-start VNF Test Cases generated some discussion among participants during the Plugtests (see Clause 10.1.1 above).

NFV-IFA013 clause 7.3.5 Update NS Operation should clarify the exact scope of “Changing the operational state of a VNF instance belonging to an NS instance”:

- Whether the operational state change to stop is intended to “shutdown” the VNFC instances
- Whether the operational state change to stop is intended to stop the service offered by the VNF Instance
- Whether both can be in scope depending on the stopType (forceful/graceful stop)

### 10.2.2 NFV-IFA007/NFV-IFA008 vs NFV-SWA001: VNF Instance States

The Network Service Update – Stop/Re-start VNF Test Cases generated some discussion among participants during the Plugtests (see Clause 10.1.1 above).

NFV-SWA001 Figure 20 summarizes VNF instance state transitions.

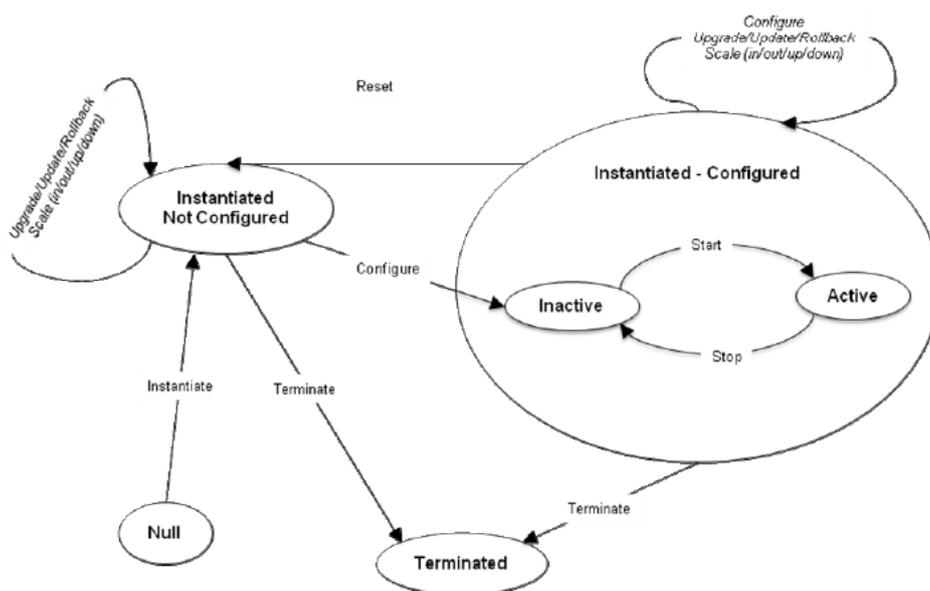


Figure 20: VNF instance state transitions

#### Figure 23 VNF instance state transitions from NFV-SWA001

NFV-IFA007/IFA008 do not provide an updated diagram for the VNF instance state transitions, but describe a number of VNF Instance states “STARTED”, “STOPPED”, “NOT\_INSTANTIATED”, “INSTANTIATED”... that do not directly map to the ones described in NFV-SWA001. This misalignment leads to a number of ambiguities and interoperability issues.

It would be desirable that ETSI NFV provided a clear statement on the applicability of NFV-SWA001 VNF instance state transitions diagram. In case of non-applicability, it would be desirable that a new diagram providing an overview of the possible VNF instance states and transitions enabled by IFA Specifications is provided.

### 10.2.3 NFV-IFA008 Ve-VNFM Interfaces

Some questions and concerns on the use of Ve-Vnfm interfaces were raised by some VNF providers. More concretely, clarifications, use cases and examples on the following would be welcome:

- 1) How VNFID and VNFCID are assigned and exchanged between MANO and VNF
- 2) Use Case for “ModifyInitialConfiguration”
- 3) Examples of VNF indicators

## 10.3 Feedback on IOP Issues

### 10.3.1 VNF management network exposure

VNFM management network exposure mechanisms (Floating IPs, VLAN exposure, ..) require alignment between VIM and MANO providers to ensure interoperability: i.e the same mechanism needs to be implemented/supported on both FUTs.

## 10.4 Other Feedback

### 10.4.1 NFVI Dimensioning and Configuration

Despite the Test Sessions having been designed to run only up to 2 VNFs at a time, the pre-testing and Plugtests experiences have showed that in order to avoid lacking resources during the Test Sessions, NFVIs should be dimensioned to be able to run a larger number of VNFs at a time. This would allow more parallel sessions to be run on the NFVI without running into resources/performance problems, to eventually complete remaining Test Sessions during low activity periods or to re-run unscheduled or pre-testing activities without impacting the scheduled sessions.

Similarly it could be advisable to dimension NFVI storage so that all the SW images could be pre-uploaded well ahead of the Test Sessions.

Also, VNF specific requirements should be exposed and taken into account for NFVI dimensioning and configuration as soon as possible in the remote integration phase.

## Annex A VNF and NS Examples

The following examples were documented during the Plugtests preparation in order to:

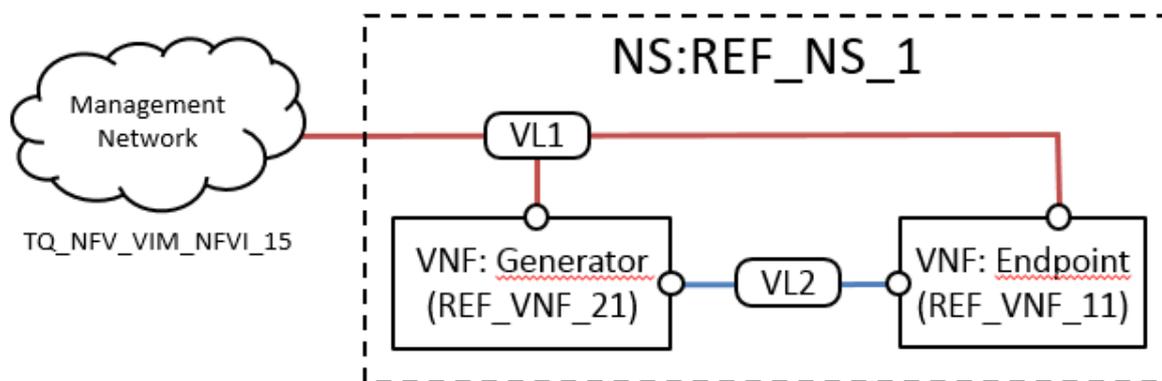
- Provide guidance to VNF providers on how to document their VNFs and reference NS in their Wiki page.
- Enable MANO providers to develop descriptor examples (NSD, VNFD) in their preferred format to be published on their Wiki page. If required, the descriptors generated by MANO providers could also include their own specific attributes.

These MANO solution specific descriptors could then be taken by VNF vendors as templates to be customized for their own VNF's requirements, and used as a starting point in their pre-testing activities with MANO providers.

### A.1 Example NS#1: Testing an endpoint VNF

The following network service captures a simple test setup where a VNF FUT is tested with a traffic generator Test VNF (or a simple VNF/VM with a basic client application).

For simplicity, this network service assumes that the VNF FUT is the endpoint of a given service (e.g. DNS, AAA, etc.) and does not require special conditions or resource allocation besides the usual in a standard cloud environments.



**Figure A.1 Example NS#1. Testing an endpoint VNF**

In this example, unless otherwise specified in the description, the following applies:

- CPs are regular para-virtualized interfaces (VirtIO or equivalent).
- VLs provide E-LAN connectivity via regular (overlay) networks provided by the VIM.
- VLs provide IP addressing via DHCP if applicable.
- Mapping between internal and external CPs may be either direct (as aliases) or via an intermediate VL.
- VIM+NFVI can guarantee predictable ordering of guest interfaces' virtual PCI addresses.

In the case of REF\_NS\_1:

- VL1 has been pre-created in the VIM, according to TQ\_NFV\_VIM\_NFVI\_15, so MANO does not need to create it. However, MANO should be made aware of its existence before the NS is created.

- Addresses given by VL1 (via DHCP) are compatible with the range allocated for VNFs in that particular NFVI+VIM. This range will be different for each NFVI+VIM participant, so it cannot be known upfront.
- DHCP in VL2 may be optional.

### A.1.1 Example VNF#11: Endpoint VNF

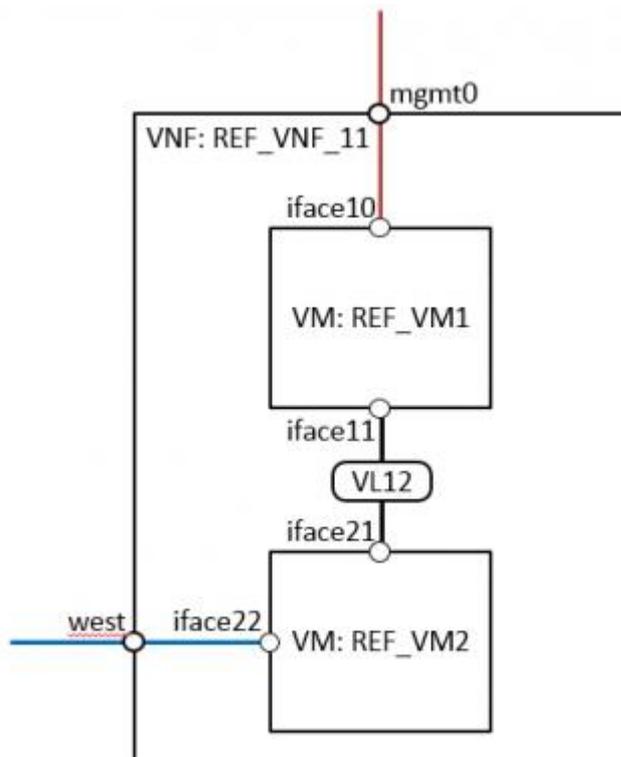


Figure A.2 Example VNF#11: Endpoint VNF

- **Name:** Ref\_VNF\_11
  - **Component:** Ref\_VM1
    - **Memory:** 2 GB
    - **CPU:** 2 vCPU
    - **Storage:** 8 GB
    - **Image:** ref\_vm1.qcow2
  - **Component:** Ref\_VM2
    - **Memory:** 4GB
    - **CPU:** 2 vCPU
    - **Storage:** 16GB
    - **Image:** ref\_vm2.qcow2
  - **Internal Virtual Link:** VL12

- No DHCP server is enabled.
- Static addressing may be used at CP iface11 and CP iface21.

### A.1.2 Example VNF#21: Generator 1 port

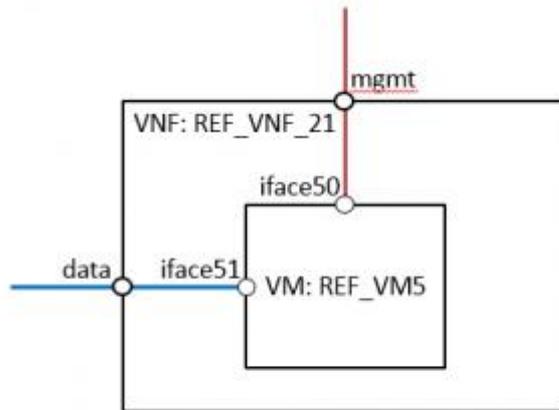
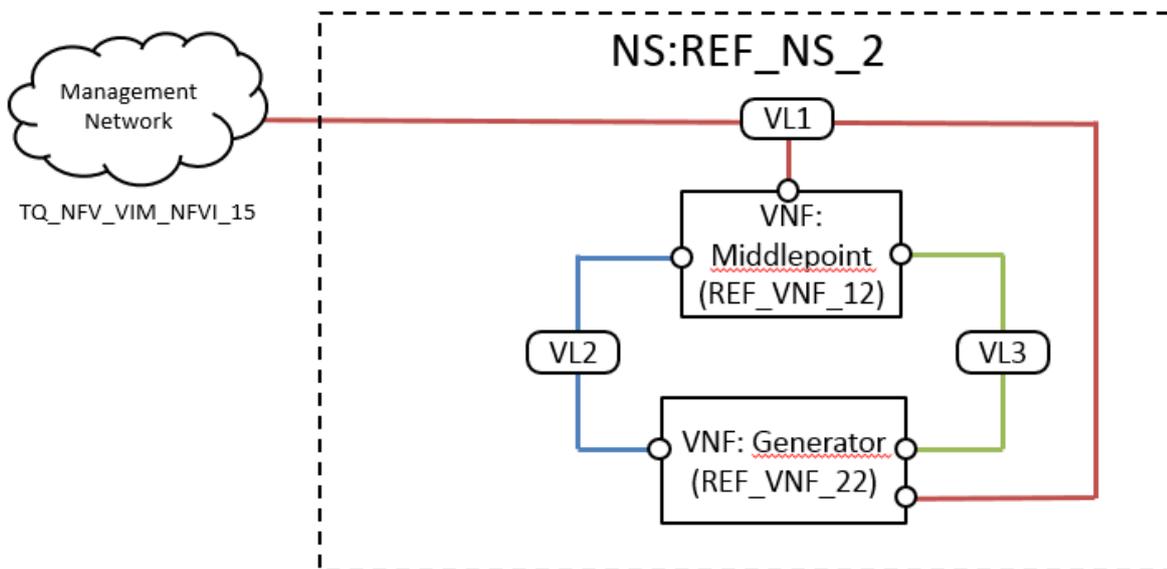


Figure A.3 Example VNF#11: Endpoint VNF

- **Name:** Ref\_VNF\_21
  - **Component:** Ref\_VM5
    - **Memory:** 1 GB
    - **CPU:** 1 vCPU
    - **Storage:** 16 GB
    - **Image:** ref\_vm21.qcow2

## A.2 Example NS #2: Testing a middle point VNF



**Figure A.4 Example NS#2: Testing a middle point VNF**

The following network service captures a more complex test setup where the VNF FUT is a middle point in the communication (e.g. router, EPC) and might require special conditions or resource allocation and connectivity foreseen in ETSI NFV specs.

In this case, the traffic generator VNF behaves as source and sink of traffic and might also require special resource allocation. Whenever these special conditions may not be available in a given MANO descriptor, fall-back options are suggested in the example to converge to conditions that would still enable interoperability with different performance characteristics.

In this example, unless otherwise specified in the description, the following applies:

- Same defaults as in NS#1
- vCPUs must be pinned to dedicated physical CPUs, with no over subscription.
  - As fall-back to complete the example (i.e. if not available), vCPUs may be assigned to oversubscribed CPUs.
- CPUs, memory and interfaces (if applicable) to be assigned to a given VM should belong to the same socket (NUMA awareness).
  - As fall-back to complete the example (i.e. if not available), NUMA awareness can be ignored.
- Memory assigned to VMs should be backed by host's huge pages memory.
  - As fall-back to complete the example (i.e. if not available), regular memory can be assigned.
- VL2 and VL3 are E-Line underlay connectivity. No DHCP is required.
  - If E-Line type not available, as fall-back to complete the example, E-Line can be emulated by E-LAN with two ports, as long as unrestricted L2 connectivity is provided (i.e. no IP or MAC)
  - If underlay connectivity is not available, as fall-back to complete the example, it can be emulated with a regular overlay (it would also require and adaptation of VNF/VM interfaces to para-virtualized).

### A.2.1 Example VNF#12: Middle point VNF

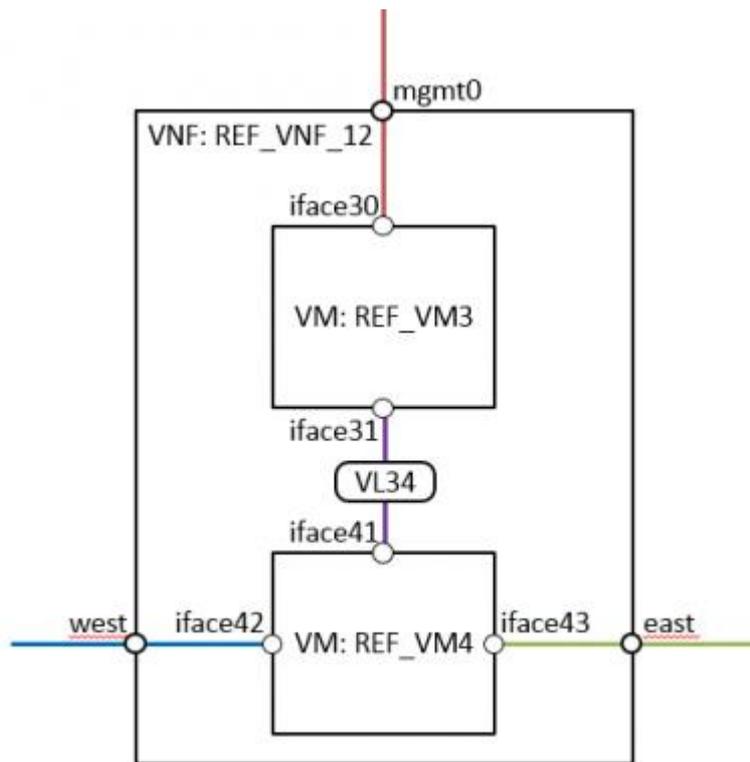


Figure A.5. Example VNF#12 Middle point VNF

- **Name:** Ref\_VNF\_12
  - **Component:** Ref\_VM3
    - **Memory:** 2 GB huge pages
    - **CPU:** 2 vCPU (= CPU)
    - **Storage:** 8 GB
    - **Image:** ref\_vm3.qcow2
  - **Component:** Ref\_VM4
    - **Memory:** 4GB
    - **CPU:** 2 vCPU
    - **Storage:** 16GB
    - **Image:** ref\_vm4.qcow2
  - **Connection Point:** iface42 (west)
    - **Type:** Passthrough
      - If passthrough type is not available, as fall-back to complete the example, SR-IOV type can be used.
      - If neither passthrough or SR-IOV types are available, as fall-back to complete the example, a para-virtualized type (e.g. Virtio) can be used.
  - **Connection Point:** iface43 (east)

- **Type:** SR-IOV
  - If SR-IOV type not available, as fall-back to complete the example, pass-through type can be used.
  - If neither pass-through nor SR-IOV types are available, as fall-back to complete the example, a para-virtualized type (e.g. Virtio) can be used.

## A.2.2 Example VNF#22: Generator 2 ports

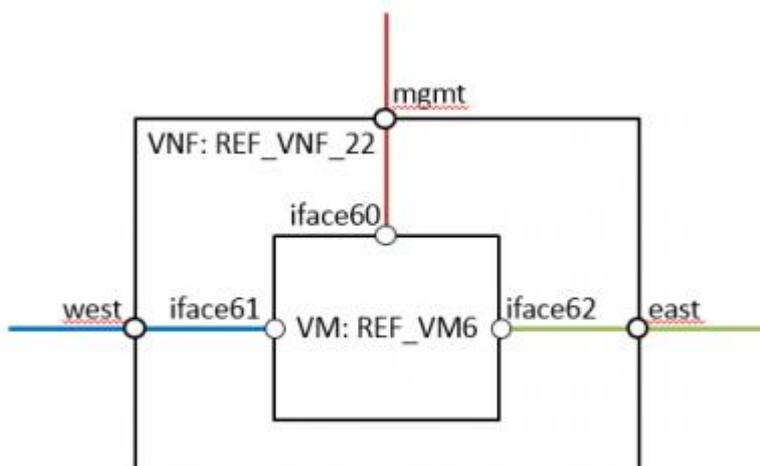


Figure A.6 Example VNF#22: Generator 2 ports

- **Name:** Ref\_VNF\_22
  - **Component:** Ref\_VM6
    - **Memory:** 1 GB huge pages
    - **CPU:** 1 vCPU (= CPU)
    - **Storage:** 16 GB
    - **Image:** ref\_vm22.qcow2
  - **Connection Point:** iface61 (west)
    - **Type:** Pass-through
      - If pass through type not available, as fall-back to complete the example, SR-IOV type can be used.
      - If neither pass-through nor SR-IOV types are available, as fall-back to complete the example, a para-virtualized type (e.g. Virtio) can be used.
  - **Connection Point:** iface62 (east)
    - **Type:** SR-IOV
      - If SR-IOV type not available, as fall-back to complete the example, pass-through type can be used.
      - If neither pass-through nor SR-IOV types are available, as fall-back to complete the example, a para-virtualized type (e.g. Virtio) can be used.

---

## History

<b>Document history</b>		
V1.0.0	15/03/2017	Publication