ETSI Plugtests Test Plan V1.0.0 (2019-07)

4th ETSI NFV Plugtests
NFV Interoperability
Sophia Antipolis, France

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

Copyright Notification

Contents

Foreword 6

Introduction 6

1 Scope 7

2 References 7
2.1 Normative references 7
2.2 Informative references 7

3 Definitions, symbols and abbreviations 9
3.1 Definitions 9
3.2 Symbols 9
3.3 Abbreviations 9

4 Test Suite Structure 10
4.1 Naming Convention 10
4.2 Test Summary 15
4.2.1 NS 15
4.2.1.1 SUT Configurations 15
4.2.1.2 Applicable Test Groups and Operations 15
4.2.3 EPA 16
4.2.3.1 SUT Configurations 16
4.2.3.2 Applicable Test Groups and Operations 16
4.2.4 SFC 17
4.2.4.1 SUT Configurations 17
4.2.4.2 Applicable Test Groups and Operations 17
4.2.5 Multi Site 18
4.2.5.1 SUT Configuration 18
4.2.5.2 Applicable Test Groups and Operations 18

5 System Under Test Configurations 19
5.1 SUT_SINGLE-VENDOR_NS 19
5.2 SUT_MULTI-VENDOR_NS 20
5.3 SUT_S-VNFM 21
5.4 SUT_AUTO-LCM-VALIDATION 22
5.5 SUT_MULTI-SITE 23

6 Interoperability Test Descriptions 24
6.1 NS 24
6.1.1 ONBOARD 24
6.1.1.1 TD_NFV_ONBOARD_VNF_PKG_001 24
6.1.1.2 TD_NFV_ONBOARD_NSD_001 24
6.1.2 INSTANTIATE 25
6.1.2.1 TD_NFV_NS_LCM_INSTANTIATE_001 25
6.1.3 SCALE NS 26
6.1.3.1 SCALE NS MANUALLY 26
6.1.3.1.1 TD_NFV_NS_LCM_SCALE_OUT_001 26
6.1.3.1.2 TD_NFV_NS_LCM_SCALE_IN_001 27
6.1.3.2 SCALE NS FROM VNF INDICATOR 28
6.1.3.2.1 TD_NFV_NS_LCM_SCALE_OUT_002a 28
6.1.3.2.2 TD_NFV_NS_LCM_SCALE_IN_002a 29
6.1.3.2.3 TD_NFV_NS_LCM_SCALE_OUT_002b 29
6.1.3.2.4 TD_NFV_NS_LCM_SCALE_IN_002b 30
6.1.3.3 SCALE NS FROM VIM PERFORMANCE METRIC 31
6.1.3.3.1 TD_NFV_NS_LCM_SCALE_OUT_003 31
6.1.3.3.2 TD_NFV_NS_LCM_SCALE_IN_003 32
6.1.4 SCALE VNF 33
6.1.4.1 SCALE VNF MANUALLY 33
6.1.4.1.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_001 33
6.1.4.1.2 TD_NFV_NS_LCM_SCALE_IN_VNF_001 34

6.1.4.2 SCALE VNF FROM VNF INDICATOR - OPTIONAL 35
6.1.4.2.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_002a 35
6.1.4.2.2 TD_NFV_NS_LCM_SCALE_IN_VNF_002a 36
6.1.4.2.3 TD_NFV_NS_LCM_SCALE_OUT_VNF_002b 37
6.1.4.2.4 TD_NFV_NS_LCM_SCALE_IN_VNF_002b 38
6.1.4.3 SCALE VNF FROM VIM PERFORMANCE METRIC - OPTIONAL 39
6.1.4.3.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_003 39
6.1.4.3.2 TD_NFV_NS_LCM_SCALE_IN_VNF_003 40
6.1.4.4 SCALE VNF FROM VNF/EM REQUEST 41
6.1.4.4.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_004 41
6.1.4.4.2 TD_NFV_NS_LCM_SCALE_IN_VNF_004 42
6.1.5 SCALE NS TO LEVEL 42
6.1.5.1 SCALE NS TO LEVEL MANUALLY 42
6.1.5.1.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_001 42
6.1.5.2 SCALE NS TO LEVEL FROM VNF INDICATOR 43
6.1.5.2.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_002 43
6.1.5.3 SCALE NS TO LEVEL FROM VIM PERFORMANCE METRIC - OPTIONAL 44
6.1.5.3.1 TD_NFV _NS_LCM_SCALE_TO_LEVEL_003 44
6.1.6 SCALE VNF TO LEVEL 46
6.1.6.1 SCALE VNF TO LEVEL MANUALLY 46
6.1.6.1.1 TD_NFV _NS_LCM_SCALE_TO_LEVEL_VNF_001 46
6.1.6.2 SCALE VNF TO LEVEL FROM VNF INDICATOR - OPTIONAL 47
6.1.6.2.1 TD_NFV _NS_LCM_SCALE_TO_LEVEL_VNF_002 47
6.1.6.3 SCALE VNF TO LEVEL FROM VIM Performance Metric - OPTIONAL 48
6.1.6.3.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_003 48
6.1.6.4 SCALE VNF TO LEVEL FROM VNF/EM REQUEST 49
6.1.6.4.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_004 49
6.1.7 UPDATE VNF 50
6.1.7.1 TD_NFV_NS_LCM_UPDATE_STOP_VNF_001 50
6.1.7.2 TD_NFV_NS_LCM_UPDATE_START_VNF_001 50
6.1.8 FAULT MANAGEMENT 51
6.1.8.1 FAULT MANAGEMENT – SUBSCRIPTION 51
6.1.8.1.1 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001 51
6.1.8.1.2 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_002 51
6.1.8.2 FAULT MANAGEMENT - NS ALARMS 52
6.1.8.2.1 TD_NFV_FM_NS_ALARM_NOTIFICATION_001 52
6.1.8.2.2 TD_NFV_FM_NS_ALARM_CLEAR_NOTIFICATION_001 53
6.1.8.2.3 TD_NFV_FM_NS_ALARM_QUERY_001 54
6.1.9 PERFORMANCE MANAGEMENT 54
6.1.9.1 TD_NFV_PM_NS_CREATE_MONITORING_JOB_001 54
6.1.9.2 TD_NFV_PM_NS_PERFORMANCE_METRICS_QUERY_001 55
6.1.9.3 TD_NFV_PM_NS_CREATE_THRESHOLD_001 56
6.1.9.4 TD_NFV_PM_NS_CREATE_SUBSCRIPTION_001 56
6.1.9.5 TD_NFV_PM_NS_CREATE_SUBSCRIPTION_002 57
6.1.9.6 TD_NFV_PM_NS_THRESHOLD_CROSSED_NOTIFICATION_001 58
6.1.9.7 TD_NFV_PM_NS_MONITORING_INFORMATION_NOTIFICATION_001 58
6.1.9.8 TD_NFV_PM_NS_DELETE_SUBSCRIPTION_002 59
6.1.9.9 TD_NFV_PM_NS_DELETE_MONITORING_JOB_001 60
6.1.9.10 TD_NFV_PM_NS_DELETE_THRESHOLD_001 60
6.1.10 VNF INDICATORS 61
6.1.10.1 TD_NFV_VNF_INDICATOR_QUERY_001 61
6.1.10.2 TD_NFV_VNF_INDICATOR_SUBSCRIPTION_001 62
6.1.10.3 TD_NFV_VNF_INDICATOR_NOTIFICATION_001 62
6.1.11 TERMINATE 63
6.1.11.1 TD_NFV_NS_LCM_TERMINATE_001 63
6.1.12 DELETE 64
6.1.12.1 TD_NFV_DELETE_NSD_001 64
6.1.12.2 TD_NFV_ DELETE_VNF_PKG_001 64
6.2 EPA 65
6.2.1 INSTANTIATE 65
6.2.1.1 TD_NFV_EPA_NS_LCM_INSTANTIATE_001 65
6.2.2 SCALE NS 66

6.2.2.1 SCALE NS MANUALLY ... 66
6.2.2.1.1 TD_NFV_EPA_NS_LCM_SCALE_OUT_001 ... 66
6.2.2.1.2 TD_NFV_EPA_NS_LCM_SCALE_IN_001 ... 67
6.2.3 SCALE VNF ... 68
6.2.3.1 SCALE VNF MANUALLY ... 68
6.2.3.1.1 TD_NFV_EPA_NS_LCM_SCALE_OUT_VNF_001 ... 68
6.2.3.1.2 TD_NFV_EPA_NS_LCM_SCALE_IN_VNF_001 ... 69
6.2.4 SCALE NS TO LEVEL ... 70
6.2.4.1 SCALE NS TO LEVEL MANUALLY ... 70
6.2.4.1.1 TD_NFV_EPA_NS_LCM_SCALE_TO_LEVEL_001 ... 70
6.2.5 SCALE VNF TO LEVEL ... 72
6.2.5.1 SCALE VNF TO LEVEL MANUALLY ... 72
6.2.5.1.1 TD_NFV _EPA_NS_LCM_SCALE_TO_LEVEL_VNF_001 ... 72
6.3 SFC ... 73
6.3.1 INSTANTIATE ... 73
6.3.1.1 TD_NFV_SFC_NS_LCM_INSTANTIATE_001 ... 73
6.4 MULTI SITE ... 74
6.4.1 INSTANTIATE ... 74
6.4.1.1 TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001 ... 74
6.4.2 SCALE NS MANUALLY ... 75
6.4.2.1 TD_NFV_MULTISITE_NS_LCM _SCALE_OUT_001 ... 75
6.4.2.2 TD_NFV_MULTISITE_NS_LCM_SCALE_IN_001 ... 76
6.4.3 SCALE VNF MANUALLY ... 77
6.4.3.1 TD_NFV_MULTISITE_NS_LCM_SCALE_OUT_VNF_001 ... 77
6.4.3.2 TD_NFV_MULTISITE_NS_LCM_SCALE_IN_VNF_001 ... 78
6.4.4 SCALE NS TO LEVEL ... 78
6.4.4.1 SCALE NS TO LEVEL MANUALLY ... 78
6.4.4.1.1 TD_NFV_MULTISITE_NS_LCM_SCALE_TO_LEVEL_001 ... 78
6.4.5 SCALE VNF TO LEVEL ... 79
6.4.5.1 SCALE VNF TO LEVEL MANUALLY ... 79
6.4.5.1.1 TD_NFV _MULTISITE_NS_LCM_SCALE_TO_LEVEL_VNF_001 ... 79
6.4.6 TERMINATE ... 80
6.4.6.1 TD_NFV_MULTISITE_NS_LCM_TERMINATE_001 ... 80

Annex A:Interoperability Feature Statements ... 82
A.1 IFS for MANO ... 82
A.2 IFS for VIM/NFVI ... 83
A.3 IFS for VNF ... 83
A.4 IFS for VNFM ... 83
A.5 IFS for NS ... 84

History ... 86

Foreword

This Test Plan has been produced by ETSI Centre for Testing and Interoperability during the preparation of the 4th ETSI NFV Plugtests.

Introduction

The present document describes the Interoperability Test Plan that was followed during the 4th ETSI NFV Plugtests held from 3rd of June to 7th of June 2019 in Sophia Antipolis, France.

The Test Plan was developed following the interoperability testing methodology and guidelines defined by ETSI NFV in [TST002] and [TST007] and building on the learninigs of the 1st , 2nd and 3rd NFV Plugtests.

1 Scope

The goal of this document is to support the interoperability test sessions run during the 4th NFV Plugtests. This event aimed at verifying early interoperability between different implementations of the main components of the NFV Architectural Framework, which included:

- Virtual Network Functions (VNF), possibly providing EM and /or specific VNF Manager (VNFM) functionality
- Management and Orchestration (MANO) solutions, providing NFV Orchestrator (NFVO) and generic VNFM functionality
- NFV Platforms providing pre-integrated NFV Infrastructure (NFVI) and Virtual Infrastructure Manager (VIM) functionality

This document includes several System Under Test Configurations to illustrate how Functions Under Test are combined to provide different end-to-end functionality

The NS compositions shown in this Test Plan are not mandated but provided as examples. The Test Descriptions have been written in such a way that also apply to more complex NS derived from the examples provided in the SUT Configurations.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative 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.

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

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[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".

[SOL002] ETSI GS NFV-SOL 002 V2.4.1: “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Ve-Vnfm Reference Point”

[SOL003] ETSI GS NFV-SOL 003 V2.4.1: “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Or-Vnfm Reference Point”

[SOL005] ETSI GS NFV-SOL 005 V2.4.1: “Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point”

[TST002] ETSI GS NFV-TST 002: “Network Functions Virtualisation (NFV); Testing Methodology; Report on NFV Interoperability Testing Methodology”

[TST007] ETSI GS NFV-TST 007: “Network Function Virtualization (NFV); Testing; Guidelines on Interoperability Testing for MANO”

3 Definitions, symbols and abbreviations

3.1 Definitions

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

3.2 Symbols

None

3.3 Abbreviations

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

4 Test Suite Structure

4.1 Naming Convention

The Test Ids of this Test Plan have been created as per the following naming convention:

TEST ID = TD_<ROOT>_<GROUP>_<SUBGROUP>_<OPERATION>_<NN>

Where <ROOT> is NFV, and <NN> corresponds to a sequential number starting at 001.

The possible values for <GROUP>, <SUBGROUP>, <OPERATION> are summarized in the tables below:

<GROUP>	<SUBGROUP>	<OPERATION>	Description
		ONBOARD_VNF_PKG	Onboard VNF Package
		ONBOARD_NSD	Onboard Network Service Descriptor
	NS_LCM	INSTANTIATE	Instantiate Network Service
		SCALE_OUT	Scale Out by adding VNF instance(s)
		SCALE_IN	Scale In by removing VNF instance(s)
		SCALE_OUT_VNF	Scale Out by adding VNFC instance(s)
		SCALE_IN_VNF	Scale In by removing VNFC instance(s)
		SCALE_TO_LEVEL	Scale To Level by changing number of VNF instance(s)
		SCALE_TO_LEVEL_VNF	Scale To Level by changing number of VNFC instance(s)
		UPDATE VNF	Start and Stop VNF instances
		TERMINATE	Terminate Network Service
	FM_NS_ALARM	SUBSCRIPTION	Subscribe and delete the fault alarm notification
		NOTIFICATION	Virtualised resource fault alarm notification propagation by NFVO
		CLEAR_NOTIFICATION	Virtualised resource fault alarm clearance notification propagation by NFVO
		QUERY	Query fault alarms
	PM_NS	CREATE_MONITORING_JOB	Monitoring NS performance metrics
		QUERY	Query NFVO to retrieve performance monitoring metrics
		CREATE_THRESHOLD	Create performance monitoring thresholds on NFVO
		CREATE_SUBSCRIPTION	Subscribe to performance metrics notifications
		THRESHOLD_CROSSED_NOTIFICATION	Expose threshold crossed notification by NFVO
		MONITORING_INFORMATION_NOTIFICATION	Expose information availability notification by the NFVO when new and updated performance metrics are available
		DELETE_SUBSCRIPTION	Delete subscription to performance monitoring notifications
		DELETE_MONITORING_JOB	Deletion of monitoring of performance metrics

		DELETE_THRESHOLD	Deletion of performance monitoring thresholds related to performance metrics
	VNF_INDICATOR	QUERY	Monitoring of VNF indicators
		SUBSCRIPTION	Subscribe to VNF indicators notifications
		NOTIFICATION	Expose VNF indicators value change notification
		DELETE_NSD	Deletion of Network Service Descriptor
		DELETE_VNF_PKG	Deletion of VNF Package

EPA	NS_LCM	INSTANTIATE	Instantiate Network Service with EPA requirements
		SCALE_OUT	Scale Out by adding VNF instance(s) with EPA requirements
		SCALE_IN	Scale In by removing VNF instance(s) with EPA requirements
		SCALE_OUT_VNF	Scale Out by adding VNFC instance(s) with EPA requirements
		SCALE_IN_VNF	Scale In by removing VNFC instance(s) with EPA requirements
		SCALE_TO_LEVEL	Scale To Level by changing number of VNF instance(s) with EPA requirements
		SCALE_TO_LEVEL_VNF	Scale To Level by changing number of VNFC instance(s) with EPA requirements

MULTISITE	NS_LCM	INSTANTIATE	Instantiate Multi Site Network Service
		SCALE_OUT	Multi Site Scale Out by adding VNF instance(s)
		SCALE_IN	Multi Site Scale In by removing VNF instance(s)
		SCALE_OUT_VNF	Multi Site Scale Out by adding VNFC instance(s)
		SCALE_IN_VNF	Multi Site Scale In by removing VNFC instance(s)
		SCALE_TO_LEVEL	Multi Site Scale To Level by changing number of VNF instance(s)
		SCALE_TO_LEVEL_VNF	Multi Site Scale To Level by changing number of VNFC instance(s)
		TERMINATE	Terminate Multi Site Network Service

4.2 Test Summary

4.2.1 NS

4.2.1.1 SUT Configurations

The System Under Test Configuration applicable to this group are:

- SUT_SINGLE-VENDOR_NS
- SUT_MULTI-VENDOR_NS
- SUT_S-VNFM
- SUT_AUTO-LCM-VALIDATION

See Clause 5 for further details

4.2.1.2 Applicable Test Groups and Operations

Group	Test IDs	# TDs
ONBOARD	TD_NFV_ONBOARD_XXX	**2**
INSTANTIATE	TD_NFV_NS_LCM_INSTANTIATE_XXX	**1**
SCALE_NS_MANUAL SCALE_NS_VNF_IND SCALE_NS_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**8**
SCALE_VNF_MANUAL SCALE_VNF_VNF_IND SCALE_VNF_VIM_METRIC SCALE_VNF_EM	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**10**
SCALE_NS_TO_LEVEL_MANUAL SCALE_NS_TO_LEVEL_VNF_IND SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**3**
SCALE_VNF_TO_LEVEL_MANUAL SCALE_VNF_TO_LEVEL_VNF_IND SCALE_VNF_TO_LEVEL_VIM_METRIC SCALE_VNF_TO_LEVEL_EM	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**4**
UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
FM_SUBSCRIPTION FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
PM_JOB PM_SUBSCRIPTION PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
VNF_IND	TD_NFV_VNF_INDICATOR_XXX	**3**
TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

4.2.3 EPA

4.2.3.1 SUT Configurations

The System Under Test Configuration applicable to this group are:

- SUT_SINGLE-VENDOR_NS
- SUT_MULTI-VENDOR_NS
- SUT_S-VNFM
- SUT_AUTO-LCM-VALIDATION

See Clause 5 for further details

4.2.3.2 Applicable Test Groups and Operations

Group	Test IDs	# TDs
ONBOARD	TD_NFV_ONBOARD_XXX	**2**
EPA_INSTANTIATE	TD_NFV_EPA_NS_LCM_INSTANTIATE_XXX	**1**
EPA_SCALE_NS_MANUAL	TD_NFV_EPA_NS_LCM_SCALE_OUT/IN_xxx	**2**
SCALE_NS_VNF_IND SCALE_NS_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**6**
EPA_SCALE_VNF_MANUAL	TD_NFV_EPA_NS_LCM_SCALE_OUT/IN_VNF_xxx	**2**
SCALE_VNF_VNF_IND SCALE_VNF_VIM_METRIC SCALE_VNF_EM	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**8**
EPA_SCALE_NS_TO_LEVEL_MANUAL	TD_NFV_EPA_NS_LCM_SCALE_TO_LEVEL_xxx	**1**
SCALE_NS_TO_LEVEL_VNF_IND SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**2**
EPA_SCALE_VNF_TO_LEVEL_MANUAL	TD_NFV_EPA_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**1**
SCALE_VNF_TO_LEVEL_MANUAL SCALE_VNF_TO_LEVEL_VNF_IND SCALE_VNF_TO_LEVEL_VIM_METRIC SCALE_VNF_TO_LEVEL_EM	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**3**
UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
FM_SUBSCRIPTION FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
PM_JOB PM_SUBSCRIPTION PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
VNF_IND	TD_NFV_VNF_INDICATOR_XXX	**3**
TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

4.2.4 SFC

4.2.4.1 SUT Configurations

The System Under Test Configuration applicable to this group are:

- SUT_SINGLE-VENDOR_NS
- SUT_MULTI-VENDOR_NS
- SUT_S-VNFM
- SUT_AUTO-LCM-VALIDATION

See Clause 5 for further details

4.2.4.2 Applicable Test Groups and Operations

Group	Test IDs	# TDs
ONBOARD	TD_NFV_ONBOARD_XXX	**2**
SFC_INSTANTIATE	TD_NFV_SFC_NS_LCM_INSTANTIATE_XXX	**1**
SCALE_NS_MANUAL SCALE_NS_VNF_IND SCALE_NS_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**8**
SCALE_VNF_MANUAL SCALE_VNF_VNF_IND SCALE_VNF_VIM_METRIC SCALE_VNF_EM	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**10**
SCALE_NS_TO_LEVEL_MANUAL SCALE_NS_TO_LEVEL_VNF_IND SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**3**
SCALE_VNF_TO_LEVEL_MANUAL SCALE_VNF_TO_LEVEL_VNF_IND SCALE_VNF_TO_LEVEL_VIM_METRIC SCALE_VNF_TO_LEVEL_EM	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**4**
UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
FM_SUBSCRIPTION FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
PM_JOB PM_SUBSCRIPTION PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
VNF_IND	TD_NFV_VNF_INDICATOR_XXX	**3**
TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

4.2.5 Multi Site

4.2.5.1 SUT Configuration

The System Under Test Configuration applicable to this group is:

- SUT_MULTI-SITE

See Clause 5 for further details

4.2.5.2 Applicable Test Groups and Operations

Group	Test IDs	# TDs
ONBOARD	TD_NFV_ONBOARD_XXX	**2**
MULTISITE_INSTANTIATE	TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_XXX	**1**
MULTISITE_SCALE_NS_MANUAL	TD_NFV_MULTISITE_NS_LCM_SCALE_OUT/IN_xxx	**2**
SCALE_NS_VNF_IND SCALE_NS_VIM_METRIC	TD_NFV_NS_LCM_SCALE_OUT/IN_xxx	**6**
MULTISITE_SCALE_VNF_MANUAL	TD_NFV_MULTISITE_NS_LCM_SCALE_OUT/IN_VNF_xxx	**2**
SCALE_VNF_VNF_IND SCALE_VNF_VIM_METRIC SCALE_VNF_EM	TD_NFV_NS_LCM_SCALE_OUT/IN_VNF_xxx	**8**
MULTISITE_SCALE_NS_TO_LEVEL_MANUAL	TD_NFV_MULTISITE_NS_LCM_SCALE_TO_LEVEL_xxx	**1**
SCALE_NS_TO_LEVEL_VNF_IND SCALE_NS_TO_LEVEL_VIM_METRIC	TD_NFV_NS_LCM_SCALE_TO_LEVEL_xxx	**2**
MULTISITE_SCALE_VNF_TO_LEVEL_MANUAL	TD_NFV_MULTISITE_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**1**
SCALE_VNF_TO_LEVEL_VNF_IND SCALE_VNF_TO_LEVEL_VIM_METRIC SCALE_VNF_TO_LEVEL_EM	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_xxx	**3**
UPDATE_VNF	TD_NFV_NS_LCM_UPDATE_XXX	**2**
FM_SUBSCRIPTION FM_ALARMS	TD_NFV_FM_NS_XXX	**5**
PM_JOB PM_SUBSCRIPTION PM_NOTIFICATION	TD_NFV_PM_ NS_XXX	**10**
VNF_IND	TD_NFV_VNF_INDICATOR_XXX	**3**
TERMINATE	TD_NFV_NS_LCM_TERMINATE_XXX	**1**
DELETE	TD_NFV_TEARDOWN_DELETE_XXX	**2**

5 System Under Test Configurations

5.1 SUT_SINGLE-VENDOR_NS

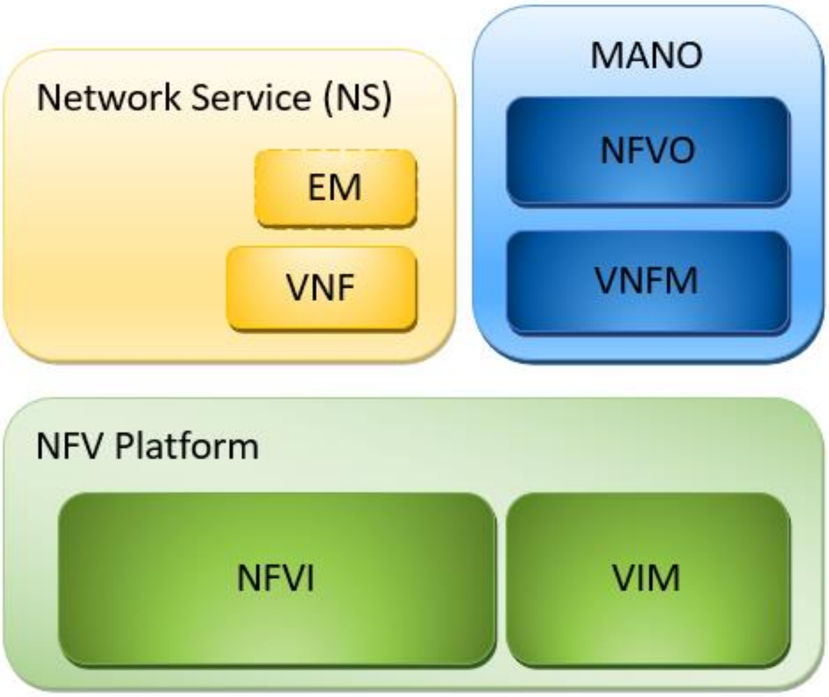

Figure 1: Single Vendor NS SUT Configuration

5.2 SUT_MULTI-VENDOR_NS

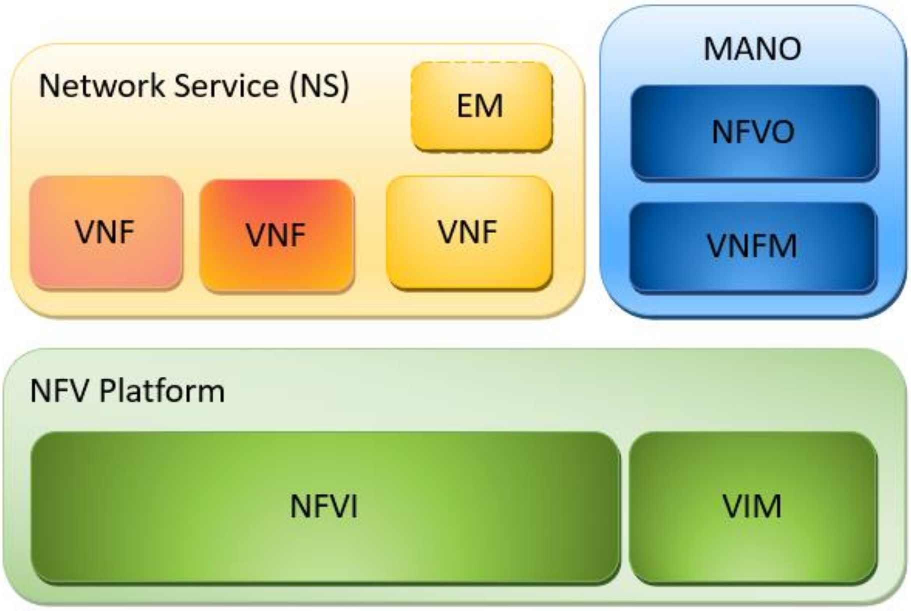

Figure 2: Multi Vendor NS SUT Configuration

5.3 SUT_S-VNFM

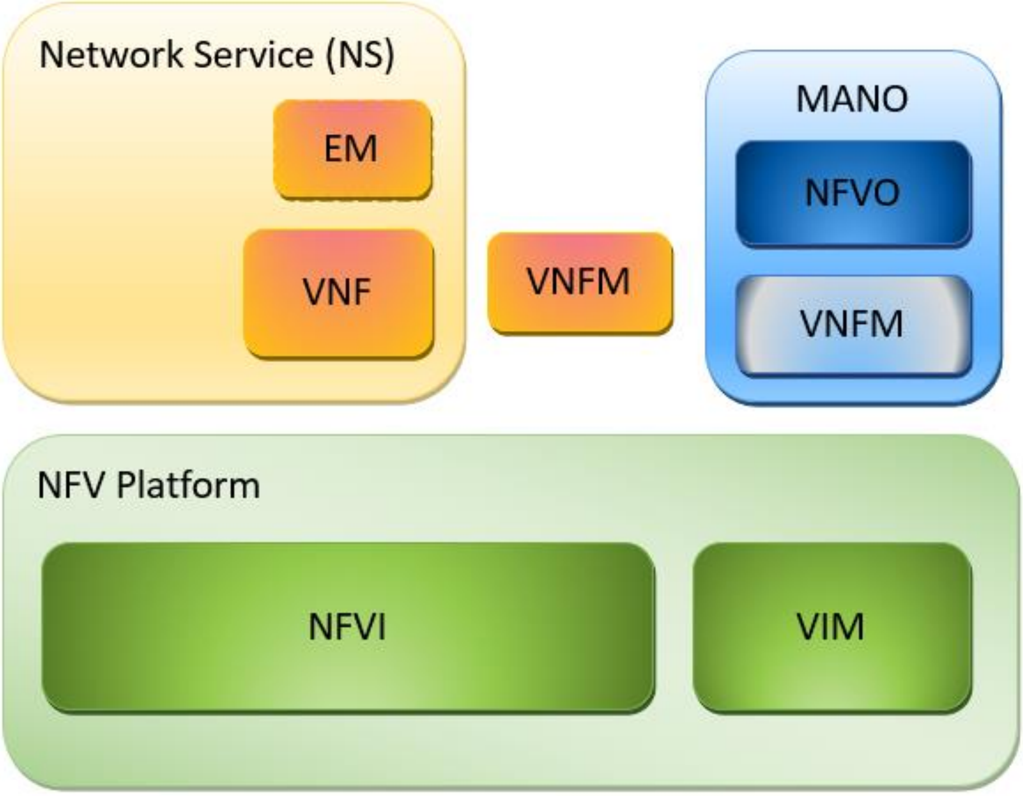

Figure 3: S-VNFM SUT Configuration

5.4 SUT_AUTO-LCM-VALIDATION

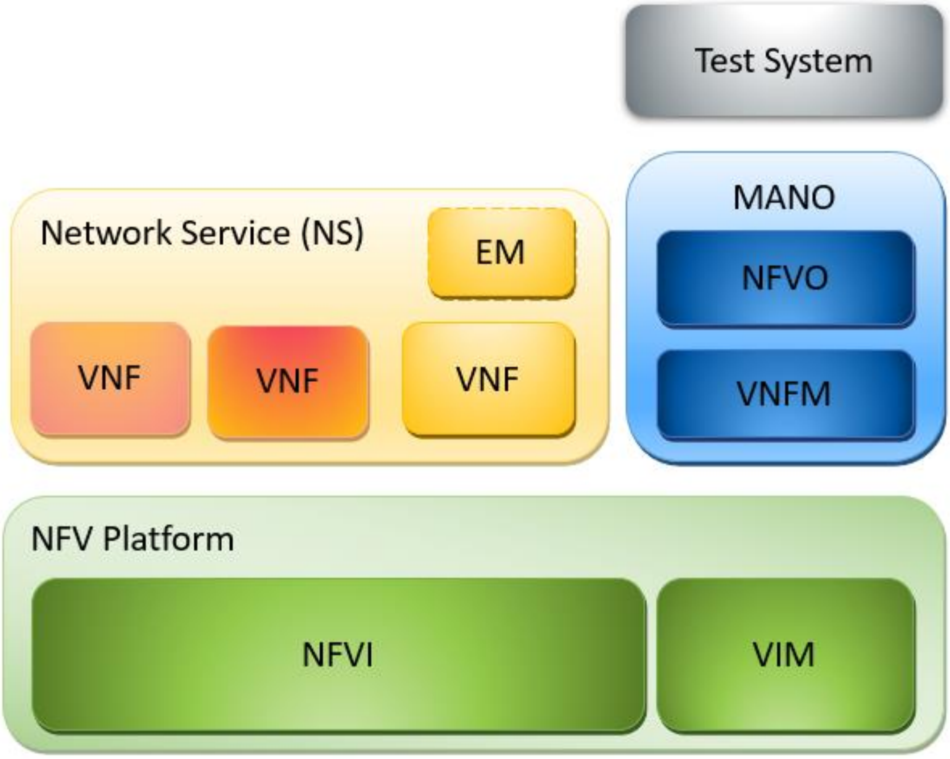

Figure 4: Auto LCM Validation SUT Configuration

5.5 SUT_MULTI-SITE

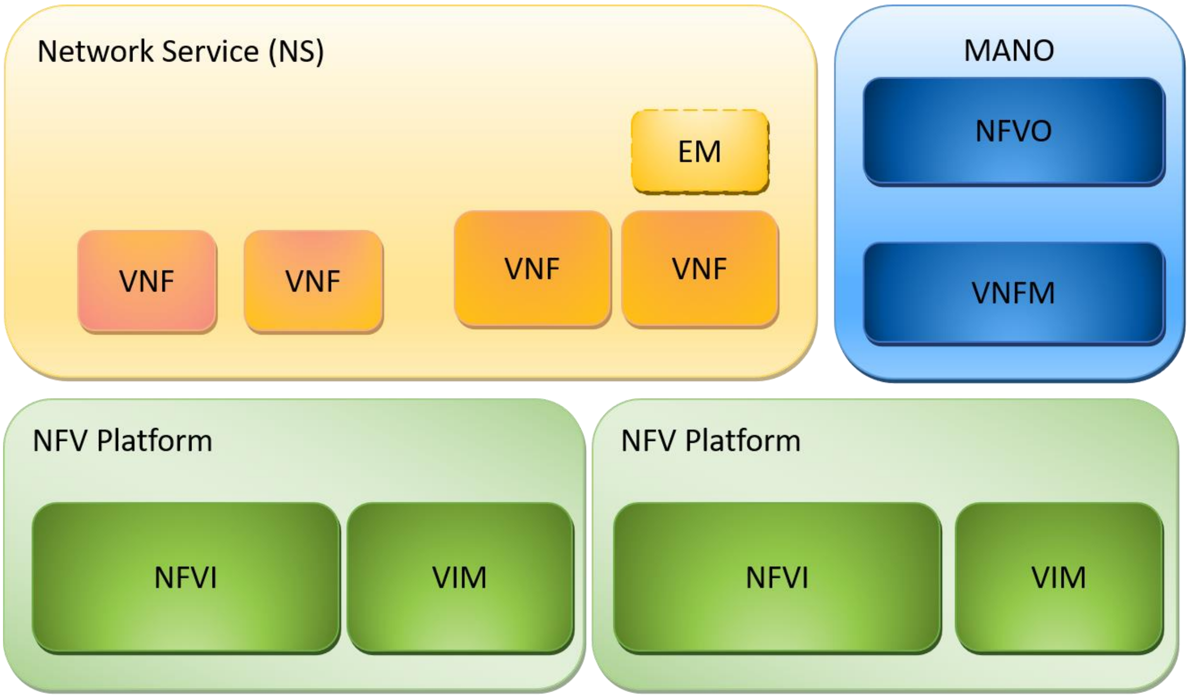

Figure 5: Multi Site SUT Configuration

6 Interoperability Test Descriptions

6.1 NS

6.1.1 ONBOARD

6.1.1.1 TD_NFV_ONBOARD_VNF_PKG_001

Interoperability Test Description				
Identifier	TD_NFV_ONBOARD_VNF_PKG_001			
Test Purpose	To on-board a VNF Package			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.7.2 [SOL005] Clause 9			
Applicability				
Pre-test conditions	* VNF Package resides on a repository reachable by NFVO * VNF Package is complete and consumable by NFVO			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the on-boarding of a VNF package in NFVO (i.e create new resource and upload content)	
	2	IOP Check	Verfiy that a new VNF package resource has been created in the NFVO	
	3	IOP Check	Verify that the content of VNF package has been successfully uploaded to the NFVO (i.e query or display VNF Package)	
	4	IOP Check	Verify that VNF package is considered ONBOARDED by the NFVO (i.e. query or display VNF package)	
IOP Verdict				

6.1.1.2 TD_NFV_ONBOARD_NSD_001

Interoperability Test Description	
Identifier	TD_NFV_ONBOARD_NSD_001
Test Purpose	To onboard a NSD
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION
References	[IFA013] Clause 7.2.2 [SOL005] Clause 5

Applicability				
Pre-test conditions	* NSD can be reached and consumed by NFVO			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the on-boarding of the NSD in the NFVO (i.e. create new resource and upload content)	
	2	IOP Check	Verify that a new NSD resource has been created in NFVO	
	3	IOP Check	Verify that the NSD content has been successfully uploaded to the NFVO (i.e. query or display NSD).	
	4	IOP Check	Verify that the NSD is considered ONBOARDED by the NFVO (i.e. query or display NSD)	
IOP Verdict				

6.1.2 INSTANTIATE

6.1.2.1 TD_NFV_NS_LCM_INSTANTIATE_001

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_INSTANTIATE_001			
Test Purpose	Verify that an NS can be successfully instantiated			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.3 [IFA007] Clause 7.2.3 [IFA006] Clauses 7.3, 7.4, 7.5 [IFA005] Clauses 7.3, 7.4, 7.5 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2			
Applicability				
Pre-test conditions	* NSD, and VNF Package(s) have been on-boarded in NFVO (TD_NFV_ONBOARD_NSD_001, TD_NFV_ONBOARD_VNF_PKG_001) * The software image is reachable by the VIM * The required resources are available on the NFVI			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS instantiation in NFVO (i.e. create new NS instance resource and instantiate it)	
	2	IOP Check	Verify that the VNFM receives instantation requests for the VNFs composing the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the VNFs composing the given NS in the VIM	

			If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the VNFs composing the given NS	
	4	IOP Check	Verify that the requested resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the VNF(s) are running and reachable through the management network	
	6	IOP Check	Verify that the initial VNF(s) configuration has been successfully applied	
	7	IOP Check	Verify that the VNF instances composing the given NS are considered INSTANTIATED by the VNFM	
	8	IOP Check	Verify that the NS instance is considered INSTANTIATED by the NFVO (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that the NS is successfully instantiated by running the end-to-end functional test	
IOP Verdict				

6.1.3 SCALE NS

6.1.3.1 SCALE NS MANUALLY

6.1.3.1.1 TD_NFV_NS_LCM_SCALE_OUT_001

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_OUT_001			
Test Purpose	To verify that a NS can be successfully scaled out (Scale_NS) by an operator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.3 [IFA007] Clause 7.2.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale out (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation request for the additional VNF(s) to be deployed for the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM	

			If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNF instance(s) are running and reachable from the management network	
	6	IOP Check	Verify that the initial configuration for the additional VNF(s) has been successfully applied	
	7	IOP Check	Verify that the additional VNF instance(s) in the NS are considered INSTANTIATED by the VNFM	
	8	IOP Check	Verify in the NFVO that the NS has been scaled out (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.3.1.2 TD_NFV_NS_LCM_SCALE_IN_001

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_IN_001			
Test Purpose	To verify that a NS can be successfully scaled in (Scale_NS) by an operator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.3 [IFA007] Clause 7.2.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [SOL005] Clause 6.3 [SOL003] Clause 5.4.8			
Applicability	* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * The current NS deployment state allows for NS scale_in operation			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale in (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the termination of the impacted VNFs	
	4	IOP Check	Verify that the impacted VNF related resources have been released in the VIM by the VNFM	
	5	IOP Check	Verify that the remaining VNF instances(s) are still running and reachable through the management network	
	6	IOP Check	Verify in the NFVO that the NS has been scaled in (i.e. query or display the NS instance resource)	

	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.3.2 SCALE NS FROM VNF INDICATOR

6.1.3.2.1 TD_NFV_NS_LCM_SCALE_OUT_002a

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_OUT_002a
Test Purpose	To verify that a NS can be successfully scaled out (Scale_NS) automatically by a VNF Indicator notification
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[IFA005] (Clause 7.3, 7.4, 7) [IFA006] (Clause 7.3, 7.4, 7.5) [IFA007] (Clause 7.2.4) [IFA008] (Clause 6.3.3) [SOL003] Clause 5.4.4 [SOL002] Clause 8.4.7, 9.4.2[
Applicability	* [IFS_NFV_NFVO_14] NFVO supports automatic NS scaling out/in triggered by VNF Indicators * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_3] NS can scale by adding/removing VNF instances (Scale_NS)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is subscribed to the targeted VNF Indicator notifications (TD_NFV_VNF_INDICATOR_NOTIFICATION_001) * NFVO is configured to trigger SCALE OUT (Scale_NS) when a given VNF Indicator value crosses a certain threshold

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF to send the VNF indicator value change notifications until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM receives instantiation request from the NFVO for the additional VNF(s) to be deployed for the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNF instance(s) are running and reachable through the management network	
	6	IOP Check	Verify that the initial configuration for the additional VNF(s) has been successfully applied	
	7	IOP Check	Verify in the NFVO that the NS has been successfully scaled out (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.3.2.2 TD_NFV_NS_LCM_SCALE_IN_002a

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_IN_002a			
Test Purpose	To verify that a NS can be successfully scaled in (Scale_NS) automatically by a VNF Indicator notification			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] (Clause 7.3, 7.4, 7.5) [IFA006] (Clause 7.3, 7.4, 7.5) [IFA007] (Clause 7.2.4) [IFA008] (Clause 6.3.3) [SOL003] Clause 5.4.4 [SOL002] Clause 8.4.7			
Applicability	* [IFS_NFV_NFVO_14] NFVO supports automatic NS scaling out/in triggered by VNF Indicators * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_3] NS can scale by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is subscribed to the targeted VNF Indicator notifications (TD_NFV_VNF_INDICATOR_NOTIFICATION_001) * NFVO is configured to trigger SCALE IN (Scale_NS) when a given VNF Indicator value crosses a certain threshold * The current NS deployment state allows for NS scale_in operation			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF to send the targeted VNF indicator notification until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the termination of the impacted VNFs	
	4	IOP Check	Verify that the impacted VNF related resources have been released in the VIM	
	5	IOP Check	Verify that the remaining VNF instances(s) are still running and reachable through the management network	
	6	IOP Check	Verify in the NFVO that the NS has been successfully scaled in (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.3.2.3 TD_NFV_NS_LCM_SCALE_OUT_002b

Interoperability Test Description

<table>
<tr><td><b>Identifier</b></td><td colspan="4">TD_NFV_NS_LCM_SCALE_OUT_002b</td></tr>
<tr><td><b>Test Purpose</b></td><td colspan="4">To verify that a NS can be successfully scaled out (Scale_NS) automatically by querying a VNF Indicator</td></tr>
<tr><td><b>Configuration</b></td><td colspan="4">SUT_SINGLE-VENDOR_NS
SUT_MULTI-VENDOR_NS
SUT_S-VNFM
SUT_MULTISITE
SUT_AUTO-LCM-VALIDATION</td></tr>
<tr><td><b>References</b></td><td colspan="4">[IFA005](Clause 7.3, 7.4, 7.5)
[IFA006] (Clause 7.3, 7.4, 7.5)
[IFA007] (Clause 7.2.4)
[IFA008] (Clause 6.3.4)
[SOL003] Clause 5.4.4
[SOL002] Clause 8.4.2, 9.4.2</td></tr>
<tr><td><b>Applicability</b></td><td colspan="4">* [IFS_NFV_NFVO_14] NFVO supports automatic NS scaling out/in triggered by VNF Indicators
* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances
* [IFS_NFV_NS_3] NS can scale by adding/removing VNF instances (Scale_NS)</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="4">* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)
* NFVO is monitoring the given VNF indicator (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001)
* NFVO is configured to trigger SCALE OUT (Scale_NS) when a given VNF Indicator value crosses a certain threshold</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td rowspan="9"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td><td><b>Result</b></td></tr>
<tr><td>1</td><td>Stimulus</td><td>In the VNF, trigger the target VNF indicator to cross the configured auto-scaling threshold value for scale out operation</td><td></td></tr>
<tr><td>2</td><td>IOP Check</td><td>Verify that the VNFM receives instantiation request from the NFVO for the additional VNF(s) to be deployed for the given NS</td><td></td></tr>
<tr><td>3</td><td>IOP Check</td><td>If VNFM is in direct mode:
• Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM
If VNFM is in indirect mode:
• Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs</td><td></td></tr>
<tr><td>4</td><td>IOP Check</td><td>Verify that the additional resources have been allocated in the VIM according to the descriptors</td><td></td></tr>
<tr><td>5</td><td>IOP Check</td><td>Verify that the additional VNF instance(s) are running and reachable through the management network</td><td></td></tr>
<tr><td>6</td><td>IOP Check</td><td>Verify that the initial configuration for the additional VNF(s) has been successfully applied</td><td></td></tr>
<tr><td>7</td><td>IOP Check</td><td>Verify in the NFVO that the NS has been successfully scaled out (i.e. query or display the NS instance resource)</td><td></td></tr>
<tr><td>8</td><td>IOP Check</td><td>Verify that NS is functional by running the end-to-end functional test</td><td></td></tr>
<tr><td><b>IOP Verdict</b></td><td colspan="3"></td><td></td></tr>
</table>

6.1.3.2.4 TD_NFV_NS_LCM_SCALE_IN_002b

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_IN_002b
Test Purpose	To verify that a NS can be successfully scaled in (Scale_NS) automatically by querying a VNF Indicator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS

	SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [IFA008] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 8.4.2			
Applicability	* [IFS_NFV_NFVO_14] NFVO supports automatic NS scaling out/in triggered by VNF Indicators * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_3] NS can scale by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is monitoring the given VNF indicator (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001) * NFVO is configured to trigger SCALE IN (Scale_NS) when a given VNF Indicator value crosses a certain threshold * The current NS deployment state allows for NS scale_in operation			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	In the VNF, trigger the target VNF indicator to cross the configured auto-scaling threshold value for scale in operation	
	2	IOP Check	Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the termination of the impacted VNFs	
	2	IOP Check	Verify that the impacted VNF related resources have been released by the VIM	
	3	IOP Check	Verify that the remaining VNF instances(s) are still running and reachable through the management network	
	4	IOP Check	Verify in the NFVO that the NS has been successfully scaled in (i.e. query or display the NS instance resource)	
	5	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.3.3 SCALE NS FROM VIM PERFORMANCE METRIC

6.1.3.3.1 TD_NFV_NS_LCM_SCALE_OUT_003

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_OUT_003
Test Purpose	To verify that a NS can be successfully scaled out (Scale_NS) automatically by a VIM Performance Metric
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE

	SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7 [IFA007] Clause 7.2.4 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving performance metrics from VNFM * [IFS_NFV_NFVO_15] NFVO supports automatic NS scaling out/in triggered by performance metrics * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is monitoring the given VIM performance metric (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001 * NFVO is configured to trigger SCALE OUT (Scale_NS) when a given VIM Performance Metric value crosses a certain threshold			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VIM to send the targeted Performance Metric until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM receives instantiation request from the NFVO for the additional VNF(s) to be deployed for the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs	
	2	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	3	IOP Check	Verify that the additional VNF instance(s) are running and reachable through the management network	
	4	IOP Check	Verify that the initial configuration for the additional VNF(s) has been successfully applied	
	5	IOP Check	Verify in the NFVO that the NS has been successfully scaled out (i.e. query or display the NS instance resource)	
	6	IOP Check	Verify that NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.3.3.2 TD_NFV_NS_LCM_SCALE_IN_003

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_IN_003
Test Purpose	To verify that a NS can be successfully scaled in (Scale_NS) automatically by a VIM Performance Metric
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7

<table>
<tr><td colspan="2"></td><td colspan="4">[IFA007] Clause 7.2
[SOL003] Clause 5.4.4</td></tr>
<tr><td colspan="2"><b>Applicability</b></td><td colspan="4">* [IFS_NFV_NFVO_11] NFVO supports receiving performance metrics from VNFM
* [IFS_NFV_NFVO_15] NFVO supports automatic NS scaling out/in triggered by performance metrics
* [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM
* [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td colspan="2"><b>Pre-test conditions</b></td><td colspan="4">* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)
* NFVO is monitoring the given VIM performance metric (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001)
* NFVO is configured to trigger SCALE IN (SCALE_NS) when a given VIM Perfromance Metric value crosses a certain threshold
* The current NS deployment state allows for NS scale_in operation</td></tr>
<tr><td colspan="6"></td></tr>
<tr><td rowspan="8"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td colspan="2"><b>Description</b></td><td><b>Result</b></td></tr>
<tr><td>1</td><td>Stimulus</td><td colspan="2">Trigger the VIM to send the targeted Performance Metric until the configured threshold is crossed</td><td></td></tr>
<tr><td>2</td><td>IOP Check</td><td colspan="2">Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS</td><td></td></tr>
<tr><td>3</td><td>IOP Check</td><td colspan="2">If VNFM is in direct mode:
• Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM
If VNFM is in indirect mode:
• Verify that the VNFM is granted by the NFVO to manage the termination of the impacted VNFs</td><td></td></tr>
<tr><td>2</td><td>IOP Check</td><td colspan="2">Verify that the impacted VNF related resources have been released in the VIM</td><td></td></tr>
<tr><td>3</td><td>IOP Check</td><td colspan="2">Verify that the remaining VNF instances(s) are still running and reachable through the management network</td><td></td></tr>
<tr><td>4</td><td>IOP Check</td><td colspan="2">Verify in the NFVO that the NS has been successfully scaled in (i.e. query or display the NS instance resource)</td><td></td></tr>
<tr><td>5</td><td>IOP Check</td><td colspan="2">Verify that NS is functional by running the end-to-end functional test</td><td></td></tr>
<tr><td><b>IOP Verdict</b></td><td colspan="4"></td><td></td></tr>
</table>

6.1.4 SCALE VNF

6.1.4.1 SCALE VNF MANUALLY

6.1.4.1.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_001

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 (Scale VNF) by an operator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION
References	[IFA013] Clause 7.3 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2

	[SOL005] Clause 6.3 [SOL003] Clause 5.4.5 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling in/out by adding/removing VNFC instances * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFC instances (Scale_VNF)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale out (Scale_VNF) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives from the NFVO scale out request for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNFC(s) are running and are reachable through the management network	
	6	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	7	IOP Check	Verify from the NFVO that the VNF in the NS has been scaled out as requested (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.1.2 TD_NFV_NS_LCM_SCALE_IN_VNF_001

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_IN_VNF_001
Test Purpose	Verify that a VNF in a NS can be successfully scaled in (Scale VNF) by an operator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION
References	[IFA013] Clause 7.3 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.5
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling in/out by adding/removing VNFC instances * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFC instances (Scale_VNF)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * The current VNF deployment state allows for scale_in operation (Scale_VNF)

Test Sequence	Step	Type	Description	Result
	1	Stimulus	Trigger NS scale in (Scale_VNF) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives from the NFVO scale in operation for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	4	IOP Check	Verify that the impacted VNFC related resources have been released in the VIM	
	5	IOP Check	Verify that the remaining VNFC(s) are still running and reachable through the management network	
	6	IOP Check	Verify from the NFVO that the VNF in the NS has been scaled in (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.2 SCALE VNF FROM VNF INDICATOR - OPTIONAL

6.1.4.2.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_002a

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_OUT_VNF_002a
Test Purpose	To verify that a VNF in a NS can be successfully scaled out (Scale_VNF) automatically by a VNF Indicator notification
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [IFA008] Clause 6.3 [SOL003] Clause 5.4.5 [SOL002] Clause 8.4.7, 9.4.2
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_15] VNFM supports automatic VNF scaling triggered by VNF indicators from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * VNFM is subscribed to the targeted VNF Indicator notifications (TD_NFV _VNF_INDICATOR_SUBSCRIPTION_001) * VNFM is configured to trigger SCALE OUT (Scale_VNF) when a given VNF Indicator value crosses a certain threshold

Test Sequence	Step	Type	Description	Result
	1	Stimulus	Trigger the VNF to send the VNF value change indicator notifications until the configured threshold is crossed	

	2	IOP Check	Verify that the VNFM starts scale out procedure for the impacted VNF	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	
	2	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	3	IOP Check	Verify that the additional VNFC(s) for the scaled VNF are running and are reachable through the management network	
	4	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	5	IOP Check	Verify from the NFVO that the VNF in the NS has been successfully scaled out (i.e. query or display the NS instance resource)	
	6	IOP Check	Verify that NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.4.2.2 TD_NFV_NS_LCM_SCALE_IN_VNF_002a

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_IN_VNF_002a			
Test Purpose	To verify that a VNF in a NS can be successfully scaled in (Scale_VNF) automatically by a VNF Indicator notification			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5) [IFA007] Clause 7.2 [IFA008] Clause 6.3 [SOL003] Clause 5.4.5 [SOL 002] Clause 8.4.7			
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_15] VNFM supports automatic VNF scaling triggered by VNF indicators from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * VNFM is subscribed to the targeted VNF Indicator notifications (TD_NFV _VNF_INDICATOR_SUBSCRIPTION_001) * VNFM is configured to trigger SCALE IN (Scale_VNF) when a given VNF Indicator value crosses a certain threshold * The current VNF deployment state allows for scale_in operation (Scale_VNF)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF to send the VNF value change indicator notifications until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM starts the scale in operation for the impacted VNF(s)	

	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	3	IOP Check	Verify that the impacted VNFC related resources have been released in the VIM	
	4	IOP Check	Verify that the remaining VNFC(s) of the scaled VNF are still running and reachable through the management network	
	5	IOP Check	Verify from the NFVO that the VNF in the NS has been successfully scaled in (i.e. query or display the NS instance resource)	
	6	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.2.3 TD_NFV_NS_LCM_SCALE_OUT_VNF_002b

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_OUT_VNF_002b			
Test Purpose	To verify that a VNF in a NS can be successfully scaled out (Scale_VNF) automatically by querying a VNF Indicator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [IFA008] Clause 6.3 [SOL003] Clause 5.4.5 [SOL002] Clause 8.4.2, 9.4.2			
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_15] VNFM supports automatic VNF scaling triggered by VNF indicators from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is monitoring the given VNF indicator (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001) * VNFM is configured to trigger SCALE OUT (Scale_VNF) when a given VNF Indicator value crosses a certain threshold			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	In the VNF, trigger the target VNF indicator to cross the configured auto-scaling threshold value for scale out operation	
	2	IOP Check	Verify that the VNFM starts scale out procedure for the impacted VNF after collecting the new VNF indicator value	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	

	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNFC(s) for the scaled VNF are running and are reachable through the management network	
	6	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	7	IOP Check	Verify from the NFVO that the VNF in the NS has been successfully scaled out (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.2.4 TD_NFV_NS_LCM_SCALE_IN_VNF_002b

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_IN_VNF_002b			
Test Purpose	To verify that a VNF in a NS can be successfully scaled in (Scale_VNF) automatically by querying a VNF Indicator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [IFA008] Clause 6.3 [SOL003] Clause 5.4.5 [SOL002] Clause 8.4.2			
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_15] VNFM supports automatic VNF scaling triggered by VNF indicators from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is monitoring the given VNF indicator (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001) * NFVO is configured to trigger SCALE IN (Scale_VNF) when a given VNF Indicator value crosses a certain threshold * The current VNF deployment state allows for scale_in operation (Scale_VNF)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	In the VNF, trigger the target VNF indicator to cross the configured auto-scaling threshold value for scale in operation	
	2	IOP Check	Verify that the VNFM starts the scale in operation for the impacted VNF after collecting the new VNF indicator value	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	4	IOP Check	Verify that the impacted VNFC related resources have been released by the VIM	

	5	IOP Check	Verify that the remaining VNFC(s) for the scaled VNF are still running and reachable through the management network	
	6	IOP Check	Verify in the NFVO that the NS has been successfully scaled in (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.3 SCALE VNF FROM VIM PERFORMANCE METRIC - OPTIONAL

6.1.4.3.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_003

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_OUT_VNF_003			
Test Purpose	To verify that a VNF in a NS can be successfully scaled out (Scale_VNF) automatically by a VIM Performance Metric			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7 [IFA007] Clause 7.2 [SOL003] Clause 5.4.5 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_18] VNFM supports automatic scaling out/in triggered by performance metrics from VIM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * VNFM is monitoring the given VIM performance metric (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001) * VNFM is configured to trigger SCALE OUT (Scale_VNF) when a given VIM Performance Metric value crosses a certain threshold			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VIM to send the targeted Performance Metric until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM starts scale out procedure for the impacted VNF after collecting the new VIM metric	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	
	2	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	

	3	IOP Check	Verify that the additional VNFC(s) are running and are reachable through the management network	
	4	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	5	IOP Check	Verify from the NFVO that the VNF in the NS has been successfully scaled out (i.e. query or display the NS instance resource)	
	6	IOP Check	Verify that NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.4.3.2 TD_NFV_NS_LCM_SCALE_IN_VNF_003

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_IN_VNF_003			
Test Purpose	To verify that a VNF in a NS can be successfully scaled in (Scale_VNF) automatically by a VIM Performance Metric			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7 [IFA007] Clause 7.2 [SOL003] Clause 5.4.5			
Applicability	* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_18] VNFM supports automatic scaling out/in triggered by performance metrics from VIM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * VNFM is monitoring the given VIM performance metric (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001) * VNFM is configured to trigger SCALE IN (Scale_VNF) when a given VIM Performance Metric value crosses a certain threshold * The current VNF deployment state allows for VNF scale_in operation (Scale_VNF)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VIM to send the targeted Performance Metric until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM starts the scale in operation for the impacted VNF after collecting the new VNF indicator value	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	3	IOP Check	Verify that the impacted VNFC related resources have been released by the VIM	
	4	IOP Check	Verify that the remaining VNFC(s) are still running and reachable through the management network	
	5	IOP Check	Verify from the NFVO that the VNF in the NS has been successfully scaled in (i.e. query or display the NS instance resource)	

	6	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.4 SCALE VNF FROM VNF/EM REQUEST

6.1.4.4.1 TD_NFV_NS_LCM_SCALE_OUT_VNF_004

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_OUT_VNF_004			
Test Purpose	To verify that a VNF in a NS can be successfully scaled out (Scale_VNF) by a request from the VNF/EM			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3.1.2, 7.4.1.2, 7.5.1.2, 7.7 [IFA006] Clause 7.3.1.2, 7.4.1.2, 7.5.1.2, 7.7 [IFA007] Clause 7.2.4 [IFA008] Clause 7.2.4 [SOL002] Clause 5.4.5, 9.4.2			
Applicability	* [IFS_NFV_VNFM_6] VNFM supports VNF scaling out/in request from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_5] VNF/EM can request scaling to VNFM			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF/EM to send a VNF scale out request to the VNFM	
	2	IOP Check	Verify that the VNFM receives from the VNF/EM a scale out request for the impacted VNF	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNFC(s) are running and are reachable through the management network	
	6	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	7	IOP Check	Verify from the NFVO that the VNF has been scaled out as requested (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.4.4.2 TD_NFV_NS_LCM_SCALE_IN_VNF_004

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_IN_VNF_004			
Test Purpose	To verify that a VNF in a NS can be successfully scaled in (Scale_VNF) by a request from the VNF/EM			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3.1.2, 7.4.1.2, 7.5.1.2, 7.7 [IFA006] Clause 7.3.1.2, 7.4.1.2, 7.5.1.2, 7.7 [IFA007] Clause 7.2.4 [IFA008] Clause 7.2.4 [SOL002] Clause 5.4.5			
Applicability	* [IFS_NFV_VNFM_6] VNFM supports VNF scaling out/in request from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_5] VNF/EM can request scaling to VNFM			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * The current VNF deployment state allows for scale_in operation (Scale_VNF)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF/EM to send a scale in (by removing VNFC instances (VMs)) request to MANO	
	2	IOP Check	Verify that the VNFM receives from the VNF/EM scale in operation for the impacted VNF	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	4	IOP Check	Verify that the impacted VNFC related resources have been released in the VIM	
	5	IOP Check	Verify that the remaining VNFC(s) are still running and reachable through the management network	
	6	IOP Check	Verify from the NFVO that the VNF in the NS has been scaled in (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.5 SCALE NS TO LEVEL

6.1.5.1 SCALE NS TO LEVEL MANUALLY

6.1.5.1.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_001

Interoperability Test Description

Identifier	TD_NFV_NS_LCM_SCALE_TO_LEVEL_001			
Test Purpose	Verify that a NS can be successfully scaled to another existing instantiation level (Scale_NS_to_level) by an operator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4, 5.4.8 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NS_11] NS supports scale to level			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * The NS initial deployment size should support scaling to or from a specified level * Current status of NS supports scale to level			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger NS scale to another existing instantiation level in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation or termination request (according to the target scale level) for the impacted VNF(s)	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete the virtualised resources for the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage instantiation or termination of the impacted VNFs	
	4	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors	
	5	IOP Check	Verify that all VNF instance(s) are running and reachable via the management network	
	6	IOP Check	Verify in the NFVO that the NS has been scaled as requested (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.5.2 SCALE NS TO LEVEL FROM VNF INDICATOR

6.1.5.2.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_002

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_SCALE_TO_LEVEL_002a

Test Purpose	Verify that a NS can be successfully scaled to another existing instantiation level (Scale_NS_to_level) automatically by a VNF indicator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [IFA008] Clause 6.3 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4, 5.4.8 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NFVO_14] NFVO supports automatic scaling triggered by VNF indicators * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM (notifications) * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS) * [IFS_NFV_NS_11] NS supports scale to level			
Pre-test conditions	* NS is instantiated (TD_NFV_SCALE-LEVEL_NS_LCM_INSTANTIATE_001) * The NS initial deployment size should support scaling to or from a specified level * NFVO is subscribed to the targeted VNF Indicator notifications (TD_NFV _VNF_INDICATOR_SUBSCRIPTION_001) * NFVO is configured to trigger "(Scale_NS_to_level)" when a given VNF indicator value crosses a certain threshold * Current status of NS supports scale to level			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger the VNF to send VNF indicator value change notifications until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM receives from the NFVO instantiation or termination request (according to the target scale level) for the impacted VNF(s)	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete the virtualised resources for the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage instantiation or termination of the impacted VNFs	
	2	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors	
	3	IOP Check	Verify that all VNF instance(s) are running and reachable via the management network	
	4	IOP Check	Verify in the NFVO that the NS has been scaled as requested (i.e. query or display the NS instance resource)	
	5	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.5.3 SCALE NS TO LEVEL FROM VIM PERFORMANCE METRIC - OPTIONAL

6.1.5.3.1 TD_NFV _NS_LCM_SCALE_TO_LEVEL_003

Interoperability Test Description

Identifier	TD_NFV_NS_LCM_SCALE_TO_LEVEL_003			
Test Purpose	Verify that a NS can be successfully scaled to another existing instantiation level (Scale_NS_to_level) automatically by a VIM Performance Metric			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4, 5.4.8			
Applicability	* [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NFVO_11] NFVO supports receiving performance metrics from VNFM (notifications) * [IFS_NFV_NFVO_15] NFVO supports automatic NS scaling out/in triggered by performance metrics * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS) * [IFS_NFV_NS_11] NS supports scale to level			
Pre-test conditions	* NS is instantiated (TD_NFV _NS_LCM_INSTANTIATE_001) *The NS initial deployment size should support scaling to or from a specified level * NFVO is monitoring the given VIM performance metric (TD_NFV _VNF_INDICATOR_SUBSCRIPTION_001) * NFVO is configured to trigger "(Scale_NS_to_level)" when a given VIM Performance metric value crosses a certain threshold * Current status of NS supports scale to level			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger the VIM to send the targeted Performance Metric until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM receives from the NFVO instantiation or termination request (according to the target scale level) for the impacted VNF(s)	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete the virtualised resources for the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage instantiation or termination of the impacted VNFs	
	2	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors	
	3	IOP Check	Verify that all VNF instance(s) are running and reachable via the management network	
	4	IOP Check	Verify in the NFVO that the NS has been scaled as requested (i.e. query or display the NS instance resource)	
	5	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.1.6 SCALE VNF TO LEVEL

6.1.6.1 SCALE VNF TO LEVEL MANUALLY

6.1.6.1.1 TD_NFV _NS_LCM_SCALE_TO_LEVEL_VNF_001

<table>
<tr><th colspan="5">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="4">TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_001</td></tr>
<tr><td><b>Test Purpose</b></td><td colspan="4">Verify that a VNF in a NS can be successfully scaled to another existing instantiation level (Scale VNF to Level) by an operator</td></tr>
<tr><td><b>Configuration</b></td><td colspan="4">SUT_SINGLE-VENDOR_NS
SUT_MULTI-VENDOR_NS
SUT_S-VNFM
SUT_MULTISITE
SUT_AUTO-LCM-VALIDATION</td></tr>
<tr><td><b>References</b></td><td colspan="4">[IFA013] Clause 7.3
[IFA005] Clause 7.3, 7.4, 7.5
[IFA006] Clause 7.3, 7.4, 7.5
[IFA007] Clause 7.2
[SOL005] Clause 6.3
[SOL003] Clause 5.4.6
[SOL002] Clause 9.4.2</td></tr>
<tr><td><b>Applicability</b></td><td colspan="4">* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances
* [IFS_NFV_NFVO_6] NFVO supports NS scale to level
* [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances
* [IFS_NFV_VNF_4] NS supports scale to level</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="4">• NS is instantiated (TD_NFV_S-VNFM-D_NS_LCM_INSTANTIATE_001)
• The NS initial deployment size should support scaling to a specified level
• NS/VNF supports scale to level by adding/removing VNFC instances</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td rowspan="8"><b>Test Sequence</b></td><td><b>Step</b></td><td></td><td><b>Description</b></td><td><b>Result</b></td></tr>
<tr><td>1</td><td>Stimulus</td><td>Trigger NS scale by scaling to another existing instantiation level a VNF in the NS in NFVO with an operator request</td><td></td></tr>
<tr><td>2</td><td>IOP Check</td><td>Verify that the VNFM receives scale to level request for the impacted VNF(s) in the given NS</td><td></td></tr>
<tr><td>3</td><td>IOP Check</td><td>If VNFM is in direct mode:
• Verify that the VNFM is granted by the NFVO to allocate or delete (according to the target scale level) the virtualised resources for scaling the impacted VNF(s)
If VNFM is in indirect mode:
• Verify that the VNFM is granted by the NFVO to manage the scale to level operation on the impacted VNF(s)</td><td></td></tr>
<tr><td>4</td><td>IOP Check</td><td>Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors by the VNFM</td><td></td></tr>
<tr><td>5</td><td>IOP Check</td><td>Verify that all VNFC instance(s) are running and reachable via the management network</td><td></td></tr>
<tr><td>6</td><td>IOP Check</td><td>Verify from the NFVO that the VNF in a NS has been successfully scaled (i.e. query or display the NS instance resource)</td><td></td></tr>
<tr><td>7</td><td>IOP Check</td><td>Verify that the NS is fuctional by running the end-to-end functional test</td><td></td></tr>
<tr><td><b>IOP Verdict</b></td><td colspan="3"></td><td></td></tr>
</table>

6.1.6.2 SCALE VNF TO LEVEL FROM VNF INDICATOR - OPTIONAL

6.1.6.2.1 TD_NFV _NS_LCM_SCALE_TO_LEVEL_VNF_002

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_002			
Test Purpose	Verify that a VNF in a NS can be successfully scaled to another existing instantiation level (Scale VNF to Level) automatically by a VNF indicator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [IFA008] Clause 6.3 [[SOL003] Clause 5.4.6 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NFVO_14] NFVO supports automatic scaling triggered by VNF indicators * [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_15] VNFM supports automatic VNF scaling triggered by VNF indicators from VNF/EM * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_4] VNF supports scale to level * [IFS_NFV_VNF_6] VNF can send VNF Indicators to VNFM (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV _NS_LCM_INSTANTIATE_001) * The VNF initial deployment size should support scaling to a specified level * VNFM is subscribed to the targeted VNF Indicator notifications (TD_NFV _VNF_INDICATOR_SUBSCRIPTION_001) * VNFM is configured to trigger "(Scale_VNF_to_level)" when a given VNF indicator value crosses a certain threshold * VNF supports scale to level by adding VNFC instances			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger the VNF to send the VNF indicator value change notifications until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM starts the scale to level operation for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete (according to the target scale level) the virtualised resources for scaling the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scale to level operation on the impacted VNF(s)	
	4	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors by the VNFM	
	5	IOP Check	Verify that all VNFC instance(s) are running and reachable via the management network	
	6	IOP Check	Verify from the NFVO that the VNF in a NS has been successfully scaled (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that the NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.6.3 SCALE VNF TO LEVEL FROM VIM Performance Metric - OPTIONAL

6.1.6.3.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_003

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_003			
Test Purpose	Verify that a VNF in a NS can be successfully scaled to another existing instantiation level (Scale VNF to Level)automatically by a VIM Performance Metric			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7 [IFA007] Clause 7.2 [[SOL003] Clause 5.4.6			
Applicability	* [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NFVO_15] NFVO supports automatic NS scaling out/in triggered by performance metrics * [IFS_NFV_VNFM_4] VNFM supports VNF scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFCs instances (Scale_VNF) * [IFS_NFV_VNF_4] VNF supports scale to level			
Pre-test conditions	• NS is instantiated (TD_NFV_SCALE-LEVEl_NS_LCM_INSTANTIATE_001) • The VNF initial deployment size should support scaling to a specified level * VNFM is monitoring the given VIM performance metric (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001) • VNFM is configured to trigger “(Scale VNF to Level)” when a given VIM Performance metric value crosses a certain threshold • NS/VNF supports scale to level by adding VNFC instances			
Test Sequence	**Step**			**Result**
	1	Stimulus	Trigger the VIM to send the targeted Performance metric until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM starts the scale to level operation for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete (according to the target scale level) the virtualised resources for scaling the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scale to level operation on the impacted VNF(s)	
	4	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors by the VNFM	
	5	IOP Check	Verify that all VNFC instance(s) are running and reachable via the management network	
	6	IOP Check	Verify from the NFVO that the VNF in a NS has been successfully scaled (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that the NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.6.4 SCALE VNF TO LEVEL FROM VNF/EM REQUEST

6.1.6.4.1 TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_004

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_SCALE_TO_LEVEL_VNF_004			
Test Purpose	Verify that a VNF in a NS can be successfully scaled to another existing instantiation level (Scale VNF to Level) by a request from VNF/EM			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clause 7.3, 7.4, 7.5, 7.7 [IFA006] Clause 7.3, 7.4, 7.5, 7.7 [IFA007] Clause 7.2 [SOL002] Clause 5.4.6, 9.4.2			
Applicability	* [IFS_NFV_VNFM_5] VNFM supports scale-to-level * [IFS_NFV_VNFM_6] VNFM supports VNF scaling out/in request from VNF/EM * [IFS_NFV_VNF_4] VNF supports scale to level * [IFS_NFV_VNF_5] VNF/EM can request scaling to VNFM			
Pre-test conditions	• NS is instantiated (TD_NFV_SCALE-LEVEl_NS_LCM_INSTANTIATE_001) • The VNF initial deployment size should support scaling to a specified level			
Test Sequence	**Step**			**Result**
	1	Stimulus	Trigger the VIM to send the targeted Performance metric until the configured threshold is crossed	
	2	IOP Check	Verify that the VNFM starts the scale to level operation for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete (according to the target scale level) the virtualised resources for scaling the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scale to level operation on the impacted VNF(s)	
	4	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors by the VNFM	
	5	IOP Check	Verify that all VNFC instance(s) are running and reachable via the management network	
	6	IOP Check	Verify from the NFVO that the VNF in a NS has been successfully scaled (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that the NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.1.7 UPDATE VNF

6.1.7.1 TD_NFV_NS_LCM_UPDATE_STOP_VNF_001

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_UPDATE_STOP_VNF_001			
Test Purpose	To verify that a VNF running in a NS can be successfully stopped			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION SUT_MULTISITE			
References	[IFA013] Clause 7.3 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.10			
Applicability	* [IFS_NFV_NFVO_7] NFVO can change VNF Operational state			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * VNF instance(s) in the NS are running			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF(s) stop operation from the NFVO	
	2	IOP Check	Verify that individual VNFC(s) inside the VNF(s) are shutdown on VIM (i.e query or display the state from VIM)	
	3	IOP Check	Verify that the VNF in the NS instance is considered STOPPED by the NFVO (i.e. query or display the NS instance resource)	
IOP Verdict				

6.1.7.2 TD_NFV_NS_LCM_UPDATE_START_VNF_001

Interoperability Test Description	
Identifier	TD_NFV_NS_LCM_UPDATE_START_VNF_001
Test Purpose	To verify that a stopped VNF in a NS can be successfully re-started
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_AUTO-LCM-VALIDATION SUT_MULTISITE
References	[IFA013] Clause 7.3 [IFA007] Clause 7.2 [SOL005] Clause 6.3
Applicability	* [IFS_NFV_NFVO_7] NFVO can change VNF Operational state
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * One VNF inside the NS has been stopped (TD_NFV_NS_LCM_UPDATE_STOP_VNF_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF(s) start operation by NFVO	
	2	IOP Check	Verify that individual VNFC(s) inside the VNF(s) are started on VIM (i.e query or display the state from VIM)	
	3	IOP Check	Verify that the VNF in NS instance is considered STARTED by the NFVO (i.e. query or display the NS instance resource)	
IOP Verdict				

6.1.8 FAULT MANAGEMENT

6.1.8.1 FAULT MANAGEMENT – SUBSCRIPTION

6.1.8.1.1 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001

Interoperability Test Description				
Identifier	TD_NFV_FM_NS_ALARM_SUBSCRIPTION_CREATE_001			
Test Purpose	Verfiy that it is possible to suscribe to alarm notifications related to faults on the NS instance			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clauses 5.3, 7.6, 8.6 [IFA013] Clauses 5.3, 7.6, 8.5 [SOL005] Clause 8.4.4 [SOL003] Clause 7.4.4			
Applicability	* [IFS_NFV_NFVO_13] NFVO supports receiving VNF faults/alarms from VNFM (notifications) * [IFS_NFV_VNFM_16] VNFM exposes VNF alarms towards NFVO (notifications) * [IFS_NFV_VNFM_14] VNFM supports receiving virtualised resource faults/alarms from VIM			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Subscribe to the alarm notifications related to NS instance faults from NFVO (i.e create new subscription resource)	
	2	IOP Check	Verify that a new subscription resource is created in the NFVO	
	3	IOP Check	Verify that the NFVO is subscribed on the VNFM to the related VNF resources faults alarm notifications	
	4	IOP Check (Optional)	If applicable, verify that the VNFM is subscribed to the related VNF or virtualised resources faults alarm notifications	
IOP Verdict				

6.1.8.1.2 TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_002

Interoperability Test Description	
Identifier	TD_NFV_FM_NS_ALARM_SUBSCRIPTION_DELETE_002

Test Purpose	Verfiy that the subscription to NS faults alarm notifications can be deleted			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[IFA005] Clauses 5.3, 7.6, 8.6 [IFA013] Clauses 5.3, 7.6, 8.5 [SOL005] Clause 8.4.4 [SOL003] Clause 7.4.4			
Applicability	* [IFS_NFV_NFVO_13] NFVO supports receiving VNF faults/alarms from VNFM (notifications) * [IFS_NFV_VNFM_16] VNFM exposes VNF alarms towards NFVO (notifications) * [IFS_NFV_VNFM_14] VNFM supports receiving virtualised resource faults/alarms from VIM			
Pre-test conditions	* NS is instantiated (TD_NFV_S-VNFM-D_NS_LCM_INSTANTIATE_001) * NFVO is subscribed to the NS faults alarm notifications (TD_NFV_FM_ALARM_SUBSCRIPTION_CREATE_001)			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Delete the subscription to related to NS instance faults in the NFVO	
	2	IOP Check	Verify that the subscription related to NS instance fault alarm notifications is deleted in the NFVO	
	3	IOP Check	Verify that the NFVO deletes the subscription on the VNFM to VNF related resources faults alarm notifications	
	4	IOP Check (Optional)	If applicable, verify that the VNFM in turn deletes its subscription to VNF or virtualised resource related resources faults alarm notifications	
IOP Verdict				

6.1.8.2 FAULT MANAGEMENT - NS ALARMS

6.1.8.2.1 TD_NFV_FM_NS_ALARM_NOTIFICATION_001

Interoperability Test Description	
Identifier	TD_NFV_ FM_NS_ALARM_NOTIFICATION_001
Test Purpose	Verify that a fault alarm notification is exposed by the NFVO when a NS related resource fails.
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[IFA005] Clauses 5.3, 7.6, 8.6 [IFA013] Clauses 5.3, 7.6, 8.5 [SOL005] Clause 8.4.6 [SOL003] Clause 7.4.6
Applicability	* [IFS_NFV_NFVO_13] NFVO supports receiving VNF faults/alarms from VNFM (notifications) * [IFS_NFV_VNFM_16] VNFM exposes VNF alarms towards NFVO (notifications) * [IFS_NFV_VNFM_14] VNFM supports receiving virtualised resource faults/alarms from VIM

Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is subscribed to the NS faults alarm notifications (TD_NFV_FM_ALARM_SUBSCRIPTION_CREATE_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger a fault on a NS related resource matching an existing alarm notification subscription	
	2	IOP Check (Optional)	If applicable, verify that a VNF fault alarm notification has been received by the VNFM (e.g. by querying the NS fault alarms database)	
	3	IOP Check	Verify that a VNF fault alarm notification is generated and dispatched by the VNFM to the NFVO	
	4	IOP Check	Verify that an NS fault alarm notification is generated and dispatched by the NFVO (e.g. by checking the GUI where applicable)	
IOP Verdict				

6.1.8.2.2 TD_NFV_FM_NS_ALARM_CLEAR_NOTIFICATION_001

Interoperability Test Description	
Identifier	TD_NFV _FM_NS_ALARM_CLEAR_NOTIFICATION_001
Test Purpose	Verify that an alarm clear notification is dispatched by the NFVO when a NS related resource failure is cleared
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[IFA005] Clauses 5.3, 7, 8 [IFA013] Clauses 5.3, 7.6, 8.5 [SOL005] Clause 8.4.6 [SOL003] Clause 7.4.6
Applicability	* [IFS_NFV_NFVO_13] NFVO supports receiving VNF faults/alarms from VNFM (notifications) * [IFS_NFV_VNFM_16] VNFM exposes VNF alarms towards NFVO (notifications) * [IFS_NFV_VNFM_14] VNFM supports receiving virtualised resource faults/alarms from VIM
Pre-test conditions	* NS is instantiated (TD_NFV_S-VNFM_NS_LCM_INSTANTIATE_001) * NS fault alarm is created on the NFVO by failing a NS related resource (TD_NFV_S-VNFM_FM_VR_ALARM_NOTIFICATION_001) * NFVO is subscribed to the NS faults alarm notifications (TD_NFV_S-VNFM_FM_ALARM_SUBSCRIPTION_CREATE_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Clear the failure of the NS related resource that triggered the alarm notification	
	2	IOP Check (Optional)	If applicable, verify that a VNF clear alarm notification has been received by the VNFM (e.g. by querying the NS fault alarms database)	
	3	IOP Check	Verify that a VNF clear alarm notification is generated and dispatched by the VNFM to the NFVO	
	4	IOP Check	Verify that an NS clear alarm notification is generated and dispatched by the NFVO (e.g. by checking the GUI where applicable)	
IOP Verdict				

6.1.8.2.3 TD_NFV_FM_NS_ALARM_QUERY_001

Interoperability Test Description	
Identifier	TD_NFV_FM_NS_ALARM_QUERY_001
Test Purpose	Verify that NS related fault alarms can be queried from the NFVO
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[IFA005] Clauses 5.3, 7, 8 [IFA013] Clauses 5.3, 7.6, 8.5 [SOL005] Clause 8.4.2, 8.4.3 [SOL003] Clause 7.4.2, 7.4.3
Applicability	* [IFS_NFV_NFVO_12] NFVO supports receiving VNF faults/alarms from VNFM (query) * [IFS_NFV_VNFM_15] VNFM exposes VNF alarms towards NFVO (query response) * [IFS_NFV_VNFM_14] VNFM supports receiving virtualised resource faults/alarms from VIM
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger a fault on a NS related resource	
	2	IOP Check (Optional)	If applicable, verify that a VNF fault alarm has been collected by the VNFM (e.g. by querying its VNF fault alarms database)	
	3	IOP Check	Verify that the VNF fault alarms are queried by NFVO from the VNFM and are properly exposed as NS fault alarms upon explicit NS fault alarms query on the NFVO (e.g. by checking the GUI where applicable)	
IOP Verdict				

6.1.9 PERFORMANCE MANAGEMENT

6.1.9.1 TD_NFV_PM_NS_CREATE_MONITORING_JOB_001

Interoperability Test Description	
Identifier	TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001
Test Purpose	Verify that performance monitoring job for monitoring NS related metrics can be created to start monitoring NS performance metrics
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.2 [SOL003] Clause 6.4.2
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)

Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * Monitoring parameters (e.g. performance metrics, metric groups) are defined in the NS descriptor (e.g. CPU usage, memory usage, etc.)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Create a performance monitoring job on the NFVO to start monitoring, with given collection and reporting periods, one or more performance metrics defined in the NS descriptor	
	2	IOP Check	If applicable, verify that the NFVO creates one or more perfomance monitoring jobs on the VNFM to collect the VNF related performance metrics	
	3	IOP Check (Optional)	If applicable, verify that the VNFM in turn issues the creation of performance monitoring jobs to collect the VNF performance metrics as virtualised resource metrics	
	4	IOP Check	Verify that performance metrics are properly collected by the VNFM with the requested collection period	
	5	IOP Check	Verify that NS performance metrics are properly collected by the NFVO from the VNFM with the requested collection period, e.g. by quering performance metrics database (if any) or checking directly the Graphical User Interface (if applicable)	
IOP Verdict				

6.1.9.2 TD_NFV_PM_NS_PERFORMANCE_METRICS_QUERY_001

Interoperability Test Description				
Identifier	TD_NFV_PM_NS_PERFORMANCE_METRICS_QUERY 001			
Test Purpose	Verify that NS related performance monitoring metrics can be retrieved in the form of reports by quering the NFVO			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[SOL005] Clause 7.4.2 [SOL003] Clause 6.4.2			
Applicability	* [IFS_NFV_NFVO_10] NFVO supports receiving VNF performance metrics from VNFM (query) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_12] VNFM exposes VNF performance metrics towards NFVO (query response)			
Pre-test conditions	* NS is instantiated (TD_NS_LCM_INSTANTIATE_001) * NS performance monitoring job is created and NFVO is colleting performance metrics (TD_NFV_PM_NS_CREATE_MONITORING_JOB_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Issue the query of one or more performance monitoring reports related to NS performance metrics to the NFVO	
	2	IOP Check	If applicable, verify that the NFVO queries VNF related perfomance monitoring reports from the VNFM to retrieve the given NS related metrics	
	3	IOP Check (Optional)	If applicable, verify that the VNFM queries virtualilsed resource related perfomance monitoring reports to retrieve the VNF related metrics	

	4	IOP Check	Verify that the performance metrics are properly retrieved from the VNFM and returned by the NFVO, e.g. checking directly the NFVO Graphical User Interface (if applicable)	
IOP Verdict				

6.1.9.3 TD_NFV_PM_NS_CREATE_THRESHOLD_001

Interoperability Test Description	
Identifier	TD_NFV_PM_NS_CREATE_THRESHOLD_001
Test Purpose	Verify that performance monitoring thresholds can be created for one or more NS related performance metrics on the NFVO
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.5, 7.4.6 [SOL003] Clause 6.4.5, 6.4.6
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring job is created and NFVO is colleting performance metrics (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Create on the NFVO a performance monitoring threshold for a NS related monitored performance metric	
	2	IOP Check	If applicable, verify that the NFVO in creates one or more thresholds on the VNFM for VNF monitored performance metrics related to the given NS instance	
	3	IOP Check (Optional)	If applicable, verify that the VNFM in turn creates one or more thresholds on virtualised resource monitored performance metrics related to the given VNF	
	4	IOP Check	Verify that the performance monitoring thresholds are properly created and maintained by the NFVO and the VNFM (e.g. checking the Graphical User Interface, if applicable)	
IOP Verdict				

6.1.9.4 TD_NFV_PM_NS_CREATE_SUBSCRIPTION_001

Interoperability Test Description	
Identifier	TD_NFV_PM_NS_CREATE_SUBSCRIPTION_001
Test Purpose	Verfiy that it is possible to suscribe to NS performance metrics notifications related to threshold crossed notifications
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION

References	[SOL005] Clause 7.4.7, 7.4.8 [SOL003] Clause 6.4.7, 6.4.8
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring threshold is created and NFVO is colleting performance metrics (TD_NFV_PM_NS_CREATE_THRESHOLD_001, TD_NFV_S-VNFM_PM_NS_CREATE_MONITORING_JOB_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Subscribe to the threshold crossed notifications related to a NS performance monitoring threshold available in the NFVO (i.e create new subscription resource)	
	2	IOP Check	Verify that a new subscription resource is created in the NFVO	
	3	IOP Check	Verify that the NFVO is subscribed on the VNFM to the threshold crossed notifications related to VNF performance metrics for the given NS instance	
	4	IOP Check (Optional)	If applicable, verify that the VNFM in turn subscribes to the threshold crossed notifications related to virtualised resource performance metrics for the given VNFs	
IOP Verdict				

6.1.9.5 TD_NFV_PM_NS_CREATE_SUBSCRIPTION_002

Interoperability Test Description	
Identifier	TD_NFV_PM_NS_CREATE_SUBSCRIPTION_002
Test Purpose	Verfiy that it is possible to suscribe to NS performance metrics notifications related to availability of perfomance information
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.7, 7.4.8 [SOL003] Clause 6.4.7, 6.4.8
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring job is created and NFVO is colleting performance metrics (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Subscribe to the performance information availability notifications related to a NS performance job in the NFVO (i.e create new subscription resource)	

	2	IOP Check	Verify that a new subscription resource is created in the NFVO	
	3	IOP Check	Verify that the NFVO is subscribed on the VNFM to the performance information availability related to VNF performance metrics for the given NS instance	
	4	IOP Check (Optional)	If applicable, verify that the VNFM in turn subscribes to the performance information availability related to virtualised resource performance metrics for the given VNFs	
IOP Verdict				

6.1.9.6 TD_NFV_PM_NS_THRESHOLD_CROSSED_NOTIFICATION_001

Interoperability Test Description				
Identifier	TD_NFV_PM_NS_THRESHOLD_CROSSED_NOTIFICATION_001			
Test Purpose	Verify that a threshold crossed notification is exposed by the NFVO when a NS performance metric crosses a previously created threshold			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[SOL005] Clause 7.4.9 [SOL003] Clause 6.4.9			
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring subscription is created to collect performance monitoring notifications related to thresholds crossed (TD_NFV_PM_ NS_CREATE SUBSCRIPTION_001)			
	Step	**Type**	**Description**	**Result**
Test Sequence	1	Stimulus	Trigger the monitored NS performance metric to cross the specified threshold	
	2	IOP Check (Optional)	If applicable, verify that the VNFM receives and collects virtualised resource threshold crossed notifications for the impacted VNF	
	3	IOP Check	Verify that a “threshold crossed” notification for the monitored VNF performance metric is generated and dispatched by the VNFM	
	4	IOP Check	Verify that a “threshold crossed” notification for the monitored NS performance metric is collected from the VNFM and dispatched by the NFVO (e.g. checking the Graphical User Interface, if applicable)	
IOP Verdict				

6.1.9.7 TD_NFV_PM_NS_MONITORING_INFORMATION_NOTIFICATION_001

Interoperability Test Description	
Identifier	TD_NFV_PM_NS_MONITORING_INFORMATION_NOTIFICATION_001
Test Purpose	Verify that a monitoring information availability notification is exposed by the NFVO when new and updated NS performance monitoring metrics are available

Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.9 [SOL003] Clause 6.4.9
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring subscription is created to collect performance monitoring notifications related to the availability of new monitoring information (TD_NFV_PM_NS_CREATE SUBSCRIPTION_002)

	Step	**Type**	**Description**	**Result**
Test Sequence	1	IOP Check (Optional)	If applicable, verify that the VNFM receives and collects virtualised resource monitoring information availability notification for the impacted VNF	
	2	IOP Check	Verify that a monitoring information availability notification for the monitored VNF performance metric is generated and dispatched by the VNFM	
	3	IOP Check	Verify that a monitoring information availability notification for the monitored NS performance metric is collected from the VNFM and dispatched by the NFVO (e.g. checking the Graphical User Interface, if applicable)	
IOP Verdict				

6.1.9.8 TD_NFV_PM_NS_DELETE_SUBSCRIPTION_002

Interoperability Test Description	
Identifier	TD_NFV_PM_NS_DELETE_SUBSCRIPTION_002
Test Purpose	Verfiy that the subscription to NS performance monitoring notifications can be deleted
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.8 [SOL003] Clause 6.4.8
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NFVO is subscribed to the NS faults alarm notifications (TD_NFV_PM_NS_CREATE_SUBSCRIPTION_001 or TD_NFV_PM_NS_CREATE_SUBSCRIPTION_002)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Delete the subscription related to NS performance monitoring notifications in the NFVO	

	2	IOP Check	Verify that the subscription related to NS performance monitoring notifications is deleted in the NFVO	
	3	IOP Check	Verify that the NFVO deletes on the VNFM the subscription to VNF related resources performance monitoring notifications	
	4	IOP Check (Optional)	If applicable, verify that the VNFM in turn deletes the subscription to virtualised resource related resources performance monitoring notifications	
IOP Verdict				

6.1.9.9 TD_NFV_PM_NS_DELETE_MONITORING_JOB_001

Interoperability Test Description	
Identifier	TD_NFV_PM_NS_DELETE_MONITORING_JOB_001
Test Purpose	Verify that performance monitoring job for monitoring NS related metrics can be deleted to stop monitoring NS performance metrics
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.3 [SOL003] Clause 5.4.3
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring job is created and NFVO is colleting performance metrics (TD_NFV_PM_ NS_CREATE_MONITORING_JOB_001)

	Step	**Type**	**Description**	**Result**
Test Sequence	1	Stimulus	Delete a performance monitoring job on the NFVO to stop monitoring one or more performance metrics	
	2	IOP Check	If applicable, verify that the NFVO deletes the related VNF perfomance monitoring jobs on the VNFM to stop monitoring the impacted VNF performance metrics	
	3	IOP Check	Verify that the NS and VNF performance metrics are no more collected by the NFVO and the VNFM, e.g. by quering the performance metrics database (if any) or checking directly the Graphical User Interface (if applicable)	
IOP Verdict				

6.1.9.10 TD_NFV_PM_NS_DELETE_THRESHOLD_001

Interoperability Test Description	
Identifier	TD_NFV _PM_NS_DELETE_THRESHOLD_001
Test Purpose	Verify that performance monitoring thresholds can be deleted for one or more NS related performance metrics on the NFVO
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM

	SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL005] Clause 7.4.6 [SOL003] Clause 6.4.6
Applicability	* [IFS_NFV_NFVO_11] NFVO supports receiving VNF performance metrics from VNFM (notifications) * [IFS_NFV_VNFM_11] VNFM supports receiving virtualised resource performance metrics from VIM * [IFS_NFV_VNFM_13] VNFM exposes VNF performance metrics towards NFVO (notifications)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * NS performance monitoring threshold is created in the NFVO (TD_NFV_PM_NS_CREATE_THRESHOLD_001)

	Step	Type	Description	Result
Test Sequence	1	Stimulus	Delete on the NFVO the performance monitoring threshold for the NS related monitored performance metric	
	2	IOP Check	If applicable, verify that the NFVO deletes the related thresholds on VNF monitored performance metrics in the VNFM	
	3	IOP Check	Verify that the NS and VNFM performance monitoring thresholds are properly deleted in the NFVO and VNFM (e.g. checking the Graphical User Interface, if applicable)	
IOP Verdict				

6.1.10 VNF INDICATORS

6.1.10.1 TD_NFV_VNF_INDICATOR_QUERY_001

Interoperability Test Description	
Identifier	TD_NFV_VNF_INDICATOR_QUERY_001
Test Purpose	To verify that a VNF indicator related to a NS instance can be monitored
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION
References	[SOL003] Clause 8.4.2, 8.4.3 [SOL002] Clause 8.4.2, 8.4.3, 8.4.4
Applicability	* [IFS_NFV_NFVO_8] NFVO supports receiving VNF indicators from VNFM (query) * [IFS_NFV_VNFM_10] VNFM exposes VNF Indicators towards NFVO (query response)
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * One or more VNF indicators are defined in the related VNF Descriptors and referenced in the NS Descriptor

	Step	Type	Description	Result
Test Sequence	1	Stimulus	Issue the query to retrieve the values of one or more VNF indicators to the NFVO	

	2	IOP Check	Verify that the NFVO queries the VNFM to retrieve the VNF indicators and collect updated indicator values	
	3	IOP Check	Verify that the VNFM queries the VNF to retrieve the indicators selected by NFVO	
	3	IOP Check	Verify that the VNF indicators values are properly retrieved and returned by the NFVO, e.g. checking directly the NFVO Graphical User Interface (if applicable)	
IOP Verdict				

6.1.10.2 TD_NFV_VNF_INDICATOR_SUBSCRIPTION_001

Interoperability Test Description				
Identifier	TD_NFV _VNF_INDICATOR_SUBSCRIPTION_001			
Test Purpose	Verfiy that it is possible to suscribe to VNF indicator notifications related to value changes			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[SOL003] Clause 8.4.5 [SOL002] Clause 8.4.5			
Applicability	* [IFS_NFV_NFVO_9] NFVO supports receiving VNF indicators from VNFM (notifications) * [IFS_NFV_VNFM_9] VNFM exposes VNF Indicators towards NFVO (notifications) * [IFS_NFV_VNFM_7] VNFM supports receiving VNF indicators from VNF/EM (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV_NS_LCM_INSTANTIATE_001) * One or more VNF indicators are defined in the related VNF Descriptors and referenced in the NS Descriptor			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Subscribe to the NFVO VNF indicator value change notifications related to indicators available in the NS description (i.e create new subscription resource)	
	2	IOP Check	Verify that a new subscription resource is created in the NFVO	
	3	IOP Check	Verify that the NFVO is susbscribed on the VNFM to the VNF indicator value change notifications for being information of actual indicator value updates	
	4	IOP Check (Optional)	If applicable, verify that the VNFM in turn subscribes on the VNF (or EM) to the VNF indicator value change notifications	
IOP Verdict				

6.1.10.3 TD_NFV_VNF_INDICATOR_NOTIFICATION_001

Interoperability Test Description	
Identifier	TD_NFV_VNF_INDICATOR_NOTIFICATION_001

Test Purpose	Verify that a VNF indicator value change notification is properly exposed by the NFVO			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTISITE SUT_AUTO-LCM-VALIDATION			
References	[SOL003] Clause 8.4.7 [SOL002] Clause 8.4.7			
Applicability	* [IFS_NFV_NFVO_9] NFVO supports receiving VNF indicators from VNFM (notifications) * [IFS_NFV_VNFM_9] VNFM exposes VNF Indicators towards NFVO (notifications) * [IFS_NFV_VNFM_7] VNFM supports receiving VNF indicators from VNF/EM (notifications)			
Pre-test conditions	* NS is instantiated (TD_NFV_S-VNFM_NS_LCM_INSTANTIATE_001) * VNF indicator subscription is created to collect indicator value change notifications (TD_NFV_S-VNFM_VNF_INDICATOR_SUBSCRIPTION_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the VNF indicator value to change (e.g. by accessing the related VNF instance and manually stimulate a VNF indicator update)	
	2	IOP Check	Verify that the VNFM receives the related VNF indicator value change notification from the VNF and dispatch it to the NFVO	
	3	IOP Check	Verify that a VNF indicator value change notification is collected from the VNFM by the NFVO (e.g. checking the Graphical User Interface, if applicable)	
IOP Verdict				

6.1.11 TERMINATE

6.1.11.1 TD_NFV_NS_LCM_TERMINATE_001

Interoperability Test Description				
Identifier	TD_NFV_NS_LCM_TERMINATE_001			
Test Purpose	To verify that a NS can be successfully terminated			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_AUTO-LCM-VALIDATION SUT-S-VNFM			
References	[IFA013] Clause 7.3.7 [SOL005] Clause 6.3.3 [IFA006] Clause 7.3.1.5, 7.4.1.5,7.5.1.5 [IFA007] Clause 7.2.7			
Applicability				
Pre-test conditions	* NS has been instantiated (TD_NFV_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS termination in NFVO	

<table>
<tr><td rowspan="4"></td><td>2</td><td>IOP Check</td><td>Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS</td><td></td></tr>
<tr><td>3</td><td>IOP Check</td><td>If VNFM is in direct mode:
• Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM
If VNFM is in indirect mode:</td><td></td></tr>
<tr><td>4</td><td>IOP Check</td><td>Verify that the resources that were allocated to the NS and VNF(s) have been released by the VIM</td><td></td></tr>
<tr><td>5</td><td>IOP Check</td><td>Verify from the NFVO that the NS instance has been terminated (i.e. query or display the state of NS instance resource)</td><td></td></tr>
<tr><td><b>IOP Verdict</b></td><td colspan="3"></td><td></td></tr>
</table>

6.1.12 DELETE

6.1.12.1 TD_NFV_DELETE_NSD_001

<table>
<tr><th colspan="5">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td colspan="4">TD_NFV_DELETE_NSD_001</td></tr>
<tr><td><b>Test Purpose</b></td><td colspan="4">To delete a NSD</td></tr>
<tr><td><b>Configuration</b></td><td colspan="4">SUT_SINGLE-VENDOR_NS
SUT_MULTI-VENDOR_NS
SUT_MULTI-SITE
SUT_AUTO-LCM-VALIDATION
SUT_S-VNFM</td></tr>
<tr><td><b>References</b></td><td colspan="4">[IFA013] Clause 7.2.6
[SOL005] Clause 5.3.5</td></tr>
<tr><td><b>Applicability</b></td><td colspan="4"></td></tr>
<tr><td colspan="5"></td></tr>
<tr><td><b>Pre-test conditions</b></td><td colspan="4">* NSD is created in NFVO (TD_NFV_ONBOARD_NSD_001)
* All NSs associated with the NSD have been terminated</td></tr>
<tr><td colspan="5"></td></tr>
<tr><td rowspan="3"><b>Test Sequence</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td><td><b>Result</b></td></tr>
<tr><td>1</td><td>Stimulus</td><td>Trigger the deletion of NSD on NFVO (i.e. delete NSD and its content)</td><td></td></tr>
<tr><td>2</td><td>IOP Check</td><td>Verify that the NSD information has been deleted in the NFVO (i.e. query or display NSD)</td><td></td></tr>
<tr><td><b>IOP Verdict</b></td><td colspan="3"></td><td></td></tr>
</table>

6.1.12.2 TD_NFV_ DELETE_VNF_PKG_001

<table>
<tr><th colspan="2">Interoperability Test Description</th></tr>
<tr><td><b>Identifier</b></td><td>TD_NFV_DELETE_VNF_PKG_001</td></tr>
<tr><td><b>Test Purpose</b></td><td>To delete a VNF Package</td></tr>
<tr><td><b>Configuration</b></td><td>SUT_SINGLE-VENDOR_NS
SUT_MULTI-VENDOR_NS
SUT_MULTI-SITE
SUT_S-VNFM</td></tr>
</table>

	SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.7.5 [SOL005] Clause 9.3			
Applicability				
Pre-test conditions	* VNF package has been on-boarded in NFVO (TD_NFV_ONBOARD_VNF_PKG _001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger the deletion of the VNF package on NFVO (i.e. delete VNF package resource and its related content)	
	2	IOP Check	Verify from the NFVO that the VNF Package information has been deleted	
IOP Verdict				

6.2 EPA

6.2.1 INSTANTIATE

6.2.1.1 TD_NFV_EPA_NS_LCM_INSTANTIATE_001

Interoperability Test Description				
Identifier	TD_NFV_EPA_NS_LCM_INSTANTIATE_001			
Test Purpose	To verify that a NS can be successfully instantiated with EPA requirements			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.3 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2.3 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_VIM_NFVI_7] VIM supports EPA			
Pre-test conditions	* NSD and VNF Package(s) have been on-boarded in NFVO (TD_NFV_ONBOARD_NSD_001, TD_NFV_ONBOARD_VNF_PKG_001) * The software image is reachable by the VIM * The required resources are available on the NFVI			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS instantiation in NFVO (i.e. create new NS instance resource and instantiate it)	

	2	IOP Check	Verify that the VNFM receives instantation requests for the VNFs composing the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the VNFs composing the given NS in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the VNFs composing the given NS	
	4	IOP Check	Verify that the requested resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that required EPA attributes have been configured as expected, e.g. checking configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough -	
	6	IOP Check	Verify that the VNF(s) are running and reachable through the management network	
	7	IOP Check	Verify that the initial VNF(s) configuration has been successfully applied	
	8	IOP Check	Verify that the VNF instances composing the given NS are considered INSTANTIATED by the VNFM	
	9	IOP Check	Verify that the NS instance is considered INSTANTIATED by the NFVO (i.e. query or display the NS instance resource)	
	10	IOP Check	Verify that the EPA requirements are matched in the NS instance and the running VNFs (e.g. performance check)	
	11	IOP Check	Verify that the NS is successfully instantiated by running the end-to-end functional test	
IOP Verdict				

6.2.2 SCALE NS

6.2.2.1 SCALE NS MANUALLY

6.2.2.1.1 TD_NFV_EPA_NS_LCM_SCALE_OUT_001

Interoperability Test Description	
Identifier	TD_NFV_EPA_NS_LCM_SCALE_OUT_001
Test Purpose	To verify that a NS can be successfully scaled out (Scale_NS) with EPA requirements by an operator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION
References	[IFA013] Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2.4 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2
Applicability	* [IFS_NFV_NFVO_14] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)

	* [IFS_NFV_VNF_9] VNF requires EPA * [IFS_NFV_VIM_NFVI_7] VIM supports EPA			
Pre-test conditions	* NS is instantiated with EPA requirements (TD_NFV_EPA_NS_LCM_INSTANTIATE_EPA_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale out (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation request for the additional VNF(s) to be deployed for the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs	
	4	IOP Check	Verify that the requested resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that required EPA attributes have been configured as expected, e.g. checking configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough -	
	6	IOP Check	Verify that the additional VNF instances(s) are running and reachable from the management network	
	7	IOP Check	Verify that the initial configuration for the additional VNF(s) has been successfully applied	
	8	IOP Check	Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check)	
	9	IOP Check	Verify that the additional VNF instance(s) in the NS are considered INSTANTIATED by the VNFM	
	10	IOP Check	Verify in the NFVO that the NS has been scaled out (i.e. query or display the NS instance resource)	
	11	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.2.2.1.2 TD_NFV_EPA_NS_LCM_SCALE_IN_001

Interoperability Test Description	
Identifier	TD_NFV_EPA_NS_LCM_SCALE_IN_001
Test Purpose	To verify that a NS can be successfully scaled in (Scale_NS) with EPA requirements by an operator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION
References	[IFA013] Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2.4 [SOL005] Clause 6.3

	[SOL003] Clause 5.4.8			
Applicability	* [IFS_NFV_NFVO_14] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS) * [IFS_NFV_VNF_9] VNF requires EPA * [IFS_NFV_VIM_NFVI_7] VIM supports EPA			
Pre-test conditions	* NS is instantiated with EPA requirements (TD_NFV_EPA_NS_LCM_INSTANTIATE_001) * The current NS deployment state allows for NS scale in operation			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale in (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the termination of the impacted VNFs	
	4	IOP Check	Verify that EPA configurations of VNF(s) to be scaled-in have been deallocated/released as expected, e.g. checking de-configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough -	
	5	IOP Check	Verify that the impacted VNF related resources have been released in the VIM by the VNFM	
	6	IOP Check	Verify that the remaining VNF instances(s) are still running and reachable through the management network	
	7	IOP Check	Verify in the NFVO that the NS has been scaled in (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.2.3 SCALE VNF

6.2.3.1 SCALE VNF MANUALLY

6.2.3.1.1 TD_NFV_EPA_NS_LCM_SCALE_OUT_VNF_001

Interoperability Test Description	
Identifier	TD_NFV_EPA_NS_LCM_SCALE_OUT_VNF_001
Test Purpose	To verify that a VNF in a NS can be successfully scaled out with EPA requirements (Scale_VNF) by an operator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION

References	[IFA013] Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2.4 [SOL005] Clause 6.3 [SOL003] Clause 5.4.5 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_14] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFC(s) instances (Scale_VNF) * [IFS_NFV_VNF_9] VNF requires EPA * [IFS_NFV_VIM_NFVI_7] VIM supports EPA			
Pre-test conditions	* NS is instantiated (TD_NFV_EPA_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale out (scale_VNF) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives from the NFVO scale out request for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	
	4	IOP Check	Verify that the requested resources have been allocated by the VIM according to the descriptors	
	5	IOP Check	Verify that required EPA attributes have been configured as expected, e.g. checking configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough -	
	6	IOP Check	Verify that the additional VNFC(s) are running and reachable from the management network	
	7	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	8	IOP Check	Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check)	
	9	IOP Check	Verify from the NFVO that the VNF in the NS has been scaled out as requested (i.e. query or display the NS instance resource)	
	10	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.2.3.1.2 TD_NFV_EPA_NS_LCM_SCALE_IN_VNF_001

Interoperability Test Description	
Identifier	TD_NFV_EPA_NS_LCM_SCALE_IN_VNF_001
Test Purpose	To verify that a VNF in a NS can be successfully scaled in (Scale_VNF) with EPA by an operator
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE

	SUT_S-VNFM SUT_AUTO-LCM-VALIDATION
References	IFA013 Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2.4 [SOL005] Clause 6.3 [SOL003] Clause 5.4.5
Applicability	* [IFS_NFV_VNFM_4] VNFM supports scaling out/in by adding/removing VNFC instances * [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFC instances * [IFS_NFV_VNF_9] VNF requires EPA * [IFS_NFV_VIM_NFVI_7] NFVI/VIM supports EPA attributes
Pre-test conditions	* NS is instantiated (TD_NFV_EPA_NS_LCM_INSTANTIATE_001) * The current NS deployment state allows for NS scale_in operation (Scale_VNF)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale in (Scale_VNF) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives from the NFVO scale in operation for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	4	IOP Check	Verify that EPA configurations of VNF(s) to be scaled-in in a NS have been deallocated/released as expected, e.g. checking de-configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough -	
	5	IOP Check	Verify that the impacted VNFCs related resources have been released by the VIM	
	6	IOP Check	Verify that the remaining VNFC(s) are still running and reachable through the management network	
	7	IOP Check	Verify from the NFVO that the VNF in the NS has been scaled in (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.2.4 SCALE NS TO LEVEL

6.2.4.1 SCALE NS TO LEVEL MANUALLY

6.2.4.1.1 TD_NFV_EPA_NS_LCM_SCALE_TO_LEVEL_001

Interoperability Test Description	
Identifier	TD_NFV_EPA_NS_LCM_SCALE_TO_LEVEL_001
Test Purpose	Verify that a NS can be successfully scaled to another existing instantiation levelwith EPA requirements (Scale_NS_to_level) by an operator

Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4, 5.4.8			
Applicability	* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_NS_11] NS supports scale to level * [IFS_NFV_VNF_9] VNF requires EPA * [IFS_NFV_VIM_NFVI_7] VIM supports EPA			
Pre-test conditions	* NS is instantiated with EPA requirements (TD_NFV_EPA_NS_LCM_INSTANTIATE_EPA_001) * The NS initial deployment size should support scaling to or from a specified level * Current status of NS supports scale to level			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger NS scale to another existing instantiation level in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation or termination request (according to the target scale level) for the impacted VNF(s)	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete the virtualised resources for the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage instantiation or termination of the impacted VNFs	
	4	IOP Check	Verify that required EPA attributes have been configured as expected, e.g. checking configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough	
	5	IOP Check	Verify that the virtualized resources have been allocated or deleted in the VIM according to the descriptors	
	6	IOP Check	Verify that all VNF instance(s) are running and reachable via the management network	
	7	IOP Check	Verify that the EPA requirements are met in the scaled VNF(s) (e.g. performance check)	
	8	IOP Check	Verify in the NFVO that the NS has been scaled as requested (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.2.5 SCALE VNF TO LEVEL

6.2.5.1 SCALE VNF TO LEVEL MANUALLY

6.2.5.1.1 TD_NFV _EPA_NS_LCM_SCALE_TO_LEVEL_VNF_001

Interoperability Test Description				
Identifier	TD_NFV _EPA_NS_LCM_SCALE_TO_LEVEL_VNF_001			
Test Purpose	Verify that a VNF in a NS can be successfully scaled to another existing instantiation level with an EPA requirements (Scale VNF to Level) by an operator			
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_MULTI-SITE SUT_S-VNFM SUT_AUTO-LCM-VALIDATION			
References	[IFA013] Clause 7.3 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.6			
Applicability	* [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_VNFM_19] VNFM supports VNFs with EPA requirements * [IFS_NFV_NS_11] NS supports scale to level * [IFS_NFV_VNF_9] VNF requires EPA * [IFS_NFV_VIM_NFVI_7] VIM supports EPA			
Pre-test conditions	• NS is instantiated (TD_NFV _NS_LCM_INSTANTIATE_001) • The NS initial deployment size should support scaling to a specified level • NS/VNF supports scale to level by adding/removing VNFC instances			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger NS scale by scaling to another existing instantiation level a VNF in the NS (Scale_VNF_to_Level) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives scale to level request for the impacted VNF(s) in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete (according to the target scale level) the virtualised resources for scaling the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scale to level operation on the impacted VNF(s)	
	4	IOP Check	Verify that required EPA attributes have been configured as expected, e.g. checking configuration of: - SR-IOV - CPU pinning - NUMA topology - PCI passthrough	
	5	IOP Check	Verify that the virtualized resources have been allocated or deleted in the VIM according to the descriptors	
	6	IOP Check	Verify that all VNFC instance(s) are running and reachable via the management network	
	7	IOP Check	Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check)	

	8	IOP Check	Verify from the NFVO that the VNF in a NS has been successfully scaled (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that the NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.3 SFC

6.3.1 INSTANTIATE

6.3.1.1 TD_NFV_SFC_NS_LCM_INSTANTIATE_001

Interoperability Test Description	
Identifier	TD_NFV_SFC_NS_LCM_INSTANTIATE_001
Test Purpose	To verify that an NS with NSH based SFC can be successfully instantiated
Configuration	SUT_SINGLE-VENDOR_NS SUT_MULTI-VENDOR_NS SUT_S-VNFM SUT_MULTI-SITE SUT_AUTO-LCM-VALIDATION
References	[IFA013] Clause 7.3 [SOL005] Clause 6.3 [IFA005] Clause 7.2 [IFA006] Clause 7.2 [IFA008] Clause 6.2 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2 IETF RFC 7665 SFC https://datatracker.ietf.org/doc/rfc7665/ IETF NSH draft https://datatracker.ietf.org/doc/rfc8300/
Applicability	* [IFS_NFV_NFVO_16] NFVO supports provisioning and configuration of network forwarding paths * [IFS_NFV_VIM_NFVI_6] NFVI/VIM supports NSH * [IFS_NFV_VNF_8] VNF supports Network Service Headers (NSH) encapsulation
Pre-test conditions	* NSD and VNF Package(s) have been on-boarded in NFVO (TD_NFV_ONBOARD_NSD_001, TD_NFV_ONBOARD_VNF_PKG_001) * The software image repository is reachable by the VIM * The required resources are available on the NFVI

	Step	**Type**	**Description**	**Result**
Test Sequence	1	Stimulus	Trigger NS instantiation in NFVO (i.e. create new NS instance resource and instantiate it)	
	2	IOP Check	Verify that the VNFM receives instantation requests for the VNFs composing the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the VNFs composing the given NS in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the VNFs composing the given NS	
	4	IOP Check	Verify that the requested resources have been allocated in the VIM according to the descriptors	
	5	IOP Check	Verify that the VNF(s) in the NS are running and reachable through the management network	

	6	IOP Check	Verify that the initial VNF(s) configuration has been successfully applied	
	7	IOP Check	Verify that the VNF instance(s) composing the given NS are considered INSTANTIATED by the VNFM	
	8	IOP Check	Verify that the NS instance is considered INSTANTIATED by the NFVO (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that the NS is functional by running the end-to-end functional test (NSH Traffic)	
IOP Verdict				

6.4 MULTI SITE

6.4.1 INSTANTIATE

6.4.1.1 TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001

Interoperability Test Description				
Identifier	TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001			
Test Purpose	To verify that an NS can be successfully instantiated across different sites			
Configuration	SUT_MULTI-SITE			
References	[IFA013] Clause 7.3 [IFA005] Clause 7.2 [IFA006] Clause 7.2 [IFA008] Clause 6.2 [IFA010] Clause 6.3 [IFA022] Clause 5.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances)			
Pre-test conditions	* NSD and VNF Package(s) have been on-boarded in NFVO (TD_NFV_ONBOARD_NSD_001, TD_NFV_ONBOARD_VNF_PKG_001) * The software image repository is reachable by the VIMs * The required resources are available on the NFVIs			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger multi-site NS instantiation in NFVO (i.e. create new NS instance resource and instantiate it)	
	2	IOP Check	Verify that the VNFM receives instantation requests for the VNFs composing the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the VNFs composing the given NS in the VIMs If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the VNFs composing the given NS	
	4	IOP Check	Verify that the requested resources have been allocated in the VIMs according to the descriptors	

	5	IOP Check	Verify that the VNF(s) have multi-site connectivity and are running and reachable through the management network	
	6	IOP Check	Verify that the initial VNF(s) configuration has been successfully applied	
	7	IOP Check	Verify that the VNF instances composing the given NS are considered INSTANTIATED by the VNFM	
	8	IOP Check	Verify that the multi-site NS instance is considered INSTANTIATED by the NFVO (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that the NS is successfully instantiated by running the end-to-end functional test	
IOP Verdict				

6.4.2 SCALE NS MANUALLY

6.4.2.1 TD_NFV_MULTISITE_NS_LCM _SCALE_OUT_001

Interoperability Test Description				
Identifier	TD_NFV_MULTISITE_NS_LCM _SCALE_OUT_001			
Test Purpose	To verify that a multi-site NS can be successfully scaled out (Scale_NS) by an operator			
Configuration	SUT_MULTI-SITE			
References	[IFA005] Clause 5.3 [IFA006] Clauses 7.3, 7.4 [IFA008] Clause 7.2 [IFA010] Clauses 6.2, 6.3 [IFA022] Clause 5.4 [IFA013] Clause 7.3.3 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances) * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* Multi-site NS is instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger multi-site NS scale out (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation request for the additional VNF(s) to be deployed for the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for the additional VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the instantiation of the additional VNFs	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors and multi-site location constraints	
	5	IOP Check	Verify that the additional VNF instances(s) have multi-site connectivity, running and reachable from the management network	

	6	IOP Check	Verify that the initial configuration for the additional VNF(s) has been successfully applied	
	7	IOP Check	Verify that the additional VNF instance(s) in the NS are considered INSTANTIATED by the VNFM	
	8	IOP Check	Verify in the NFVO that the multi-site NS has been scaled out (i.e. query or display the NS instance resource)	
	9	IOP Check	Verify that multi-site NS is functional by running the end-to-end functional test	
IOP Verdict				

6.4.2.2 TD_NFV_MULTISITE_NS_LCM_SCALE_IN_001

Interoperability Test Description				
Identifier	TD_NFV_MULTISITE_NS_LCM_SCALE_IN_001			
Test Purpose	To verify that a multi-site NS can be successfully scaled in (Scale_NS) by an operator			
Configuration	SUT_MULTI-SITE			
References	[IFA005] Clause 5.3 [IFA006] Clauses 7.3, 7.4 [IFA008] Clause 7.2 [IFA010] Clauses 6.2, 6.3 [IFA022] Clause 5.4 [IFA013] Clause 7.3.3 [SOL005] Clause 6.3 [SOL003] Clause 5.4.8			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances) * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NS_10] NS can scale out/in by adding/removing VNF instances (Scale_NS)			
Pre-test conditions	* Multi-site NS is instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001) * The current NS deployment state allows for NS scale_in operation			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger multi-site NS scale in (Scale_NS) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the termination of the impacted VNFs	
	4	IOP Check	Verify that the impacted VNF related resources have been released in the proper VIM site / VIM instance	
	5	IOP Check	Verify that the remaining VNF instances(s) are still running and reachable through the management network	
	6	IOP Check	Verify in the NFVO that the multi-site NS has been scaled in (i.e. query or display the NS instance resource)	

	7	IOP Check	Verify that multi-site NS is functional by running the end-to-end functional test	
IOP Verdict				

6.4.3 SCALE VNF MANUALLY

6.4.3.1 TD_NFV_MULTISITE_NS_LCM_SCALE_OUT_VNF_001

Interoperability Test Description				
Identifier	TD_NFV_MULTISITE_NS_LCM_SCALE_OUT_VNF_001			
Test Purpose	To verify that a VNF in a multi-site NS can be successfully scaled out (Scale_VNF) by an operator			
Configuration	SUT_MULTI-SITE			
References	[IFA005] Clause 5.3 [IFA006] Clauses 7.3, 7.4 [IFA013] Clause 7.3 [IFA008] Clause 7.2 [IFA010] Clauses 6.2, 6.3 [IFA022] Clause 5.4 [SOL005] Clause 6.3 [SOL003] Clause 5.4.5, 5.4.4 [SOL002] Clause 9.4.2			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances) * [IFS_NFV_VNFM_4] VNFM supports VNF scaling in/out by adding/removing VNFC instances * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFC instances (Scale_VNF)			
Pre-test conditions	* Multi-Site NS is instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger multi-site NS scale out (Scale_VNF) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives from the NFVO scale out request for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate the virtualised resources required for scaling the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling out of the impacted VNF(s)	
	4	IOP Check	Verify that the additional resources have been allocated in the VIM according to the descriptors and multi-site location constraints	
	5	IOP Check	Verify that the additional VNFC instances(s) have multi-site connectivity,running and reachable from the management network	
	6	IOP Check	Verify that the initial configuration for the additional VNFC(s) has been successfully applied	
	7	IOP Check	Verify in the NFVO that the VNF in multi-site NS has been scaled out as requested (i.e. query or display the NS instance resource)	
	8	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.4.3.2 TD_NFV_MULTISITE_NS_LCM_SCALE_IN_VNF_001

Interoperability Test Description				
Identifier	TD_NFV_MULTISITE_NS_LCM_SCALE_IN_VNF_001			
Test Purpose	To verify that a VNF in a multi-site NS can be successfully scaled in (Scale_VNF) by an operator			
Configuration	SUT_MULTI-SITE			
References	[IFA005 Clause 5.3 [IFA006 Clauses 7.3, 7.4 [IFA013 Clause 7.3 [IFA008 Clause 7.2 [IFA010 Clauses 6.2, 6.3 [IFA022 Clause 5.4 [SOL005] Clause 6.3 [SOL003] Clause 5.4.5			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances) *[IFS_NFV_VNFM_4] VNFM supports VNF scaling in/out by adding/removing VNFC instances * [IFS_NFV_VNF_3] VNF can scale out/in by adding/removing VNFC instances (Scale_VNF)			
Pre-test conditions	* Multi-site NS is instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001) * The current VNF deployment state allows for scale_in operation (Scale_VNF)			
Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger NS scale in (Scale_VNF) in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives from the NFVO scale in operation for the impacted VNF in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources related to the impacted VNF If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scaling in of the impacted VNF(s)	
	4	IOP Check	Verify that the impacted VNFC related resources have been released by the proper VIM site / VIM instance	
	5	IOP Check	Verify that the remaining VNFC instances(s) have multi-site connectivity and are still running and reachable through the management network	
	6	IOP Check	Verify from the NFVO that the VNF in the NS has been scaled in (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that multi-site NS is functional by running the end-to-end functional test	
IOP Verdict				

6.4.4 SCALE NS TO LEVEL

6.4.4.1 SCALE NS TO LEVEL MANUALLY

6.4.4.1.1 TD_NFV_MULTISITE_NS_LCM_SCALE_TO_LEVEL_001

Interoperability Test Description	
Identifier	TD_NFV_MULTISITE_NS_LCM_SCALE_TO_LEVEL_001

Test Purpose	Verify that a multi-site NS can be successfully scaled to another existing instantiation level (Scale_NS_to_level) by an operator			
Configuration	SUT_MULTI-SITE			
References	[IFA013] Clause 7.3.4 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5 [IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.4, 5.4.8			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances) * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NS_11] NS supports scale to level			
Pre-test conditions	* Multi-site NS is instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001) * The NS initial deployment size should support scaling to or from a specified level * Current status of NS supports scale to level			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger multi-site NS scale to another existing instantiation level in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives instantiation or termination request (according to the target scale level) for the impacted VNF(s)	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete the virtualised resources for the impacted VNFs in the VIM If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage instantiation or termination of the impacted VNFs	
	4	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors	
	5	IOP Check	Verify that all VNF instance(s) are running and reachable via the management network	
	6	IOP Check	Verify in the NFVO that the multi-site NS has been scaled as requested (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that NS is functional by running the end-to-end functional test	
IOP Verdict				

6.4.5 SCALE VNF TO LEVEL

6.4.5.1 SCALE VNF TO LEVEL MANUALLY

6.4.5.1.1 TD_NFV _MULTISITE_NS_LCM_SCALE_TO_LEVEL_VNF_001

Interoperability Test Description	
Identifier	TD_NFV_MULTISITE_NS_LCM_SCALE_TO_LEVEL_VNF_001
Test Purpose	Verify that a VNF in a multi-site NS can be successfully scaled to another existing instantiation level (Scale VNF to Level) by an operator
Configuration	SUT_MULTI-SITE
References	[IFA013] Clause 7.3 [IFA005] Clause 7.3, 7.4, 7.5 [IFA006] Clause 7.3, 7.4, 7.5

	[IFA007] Clause 7.2 [SOL005] Clause 6.3 [SOL003] Clause 5.4.6			
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances) * [IFS_NFV_NFVO_5] NFVO supports NS scaling by adding/removing VNF instances * [IFS_NFV_NFVO_6] NFVO supports NS scale to level * [IFS_NFV_NS_4] VNF(s) in the NS can scale to level			
Pre-test conditions	* Multisite NS is instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001) * The NS initial deployment size should support scaling to a specified level * NS/VNF supports scale to level by adding/removing VNFC instances			
Test Sequence	**Step**		**Description**	**Result**
	1	Stimulus	Trigger multi-site NS scale by scaling to another existing instantiation level a VNF in the NS in NFVO with an operator request	
	2	IOP Check	Verify that the VNFM receives scale to level request for the impacted VNF(s) in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to allocate or delete (according to the target scale level) the virtualised resources for scaling the impacted VNF(s) If VNFM is in indirect mode: • Verify that the VNFM is granted by the NFVO to manage the scale to level operation on the impacted VNF(s)	
	4	IOP Check	Verify that the virtualised resources have been allocated or deleted in the VIM according to the descriptors by the VNFM	
	5	IOP Check	Verify that all VNFC instance(s) are running and reachable via the management network	
	6	IOP Check	Verify from the NFVO that the VNF in a multi-site NS has been successfully scaled (i.e. query or display the NS instance resource)	
	7	IOP Check	Verify that the NS is fuctional by running the end-to-end functional test	
IOP Verdict				

6.4.6 TERMINATE

6.4.6.1 TD_NFV_MULTISITE_NS_LCM_TERMINATE_001

Interoperability Test Description	
Identifier	TD_NFV_MULTISITE_NS_LCM_TERMINATE_001
Test Purpose	To verify that a Multi Site NS can be successfully terminated
Configuration	SUT_MULTI-SITE
References	[IFA013] Clause 7.3 [IFA005] Clause 7.3, 7.4, 7.5 [IFA008] Clause 7.2 [SOL005] Clause 6.3
Applicability	* [IFS_NFV_NFVO_4] NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances)
Pre-test conditions	* Multi Site NS has been instantiated (TD_NFV_MULTISITE_NS_LCM_INSTANTIATE_001)

Test Sequence	**Step**	**Type**	**Description**	**Result**
	1	Stimulus	Trigger Multi Site NS termination in NFVO	
	2	IOP Check	Verify that the VNFM receives terminate request from the NFVO for the VNF(s) to be removed in the given NS	
	3	IOP Check	If VNFM is in direct mode: • Verify that the VNFM is granted by the NFVO to delete the virtualised resources in use by the impacted VNFs in the VIM If VNFM is in indirect mode:	
	4	IOP Check	Verify that the resources that were allocated to the Multi Site NS and VNF(s) have been released by the involved VIMs	
	5	IOP Check	Verify from the NFVO that the multi-site NS instance has been terminated (i.e. query or display the state of NS instance resource)	
IOP Verdict				

Annex A:Interoperability Feature Statements

A.1 IFS for MANO

IFS_ID	Description	Support
[IFS_NFV_NFVO_1]	NFVO provides generic VNFM functionality	
[IFS_NFV_NFVO_2]	NFVO supports specific VNFMs (external) in direct mode (resource management by VNFM)	
[IFS_NFV_NFVO_3]	NFVO supports specific VNFMs (external) in in-direct mode (resource management by MANO)	
[IFS_NFV_NFVO_4]	NFVO supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances)	
[IFS_NFV_NFVO_5]	NFVO supports NS scaling out/in by adding/removing VNF instances	
[IFS_NFV_NFVO_6]	NFVO supports NS scale to level	
[IFS_NFV_NFVO_7]	NFVO can change VNF Operational state	
[IFS_NFV_NFVO_8]	NFVO supports receiving VNF indicators from VNFM (query)	
[IFS_NFV_NFVO_9]	NFVO supports receiving VNF indicators from VNFM (notifications)	
[IFS_NFV_NFVO_10]	NFVO supports receiving VNF performance metrics from VNFM (query)	
[IFS_NFV_NFVO_11]	NFVO supports receiving VNF performance metrics from VNFM (notifications)	
[IFS_NFV_NFVO_12]	NFVO supports receiving VNF faults/alarms from VNFM (query)	
[IFS_NFV_NFVO_13]	NFVO supports receiving VNF faults/alarms from VNFM (notifications)	
[IFS_NFV_NFVO_14]	NFVO supports automatic NS scaling out/in triggered by VNF Indicators	
[IFS_NFV_NFVO_15]	NFVO supports automatic NS scaling out/in triggered by performance metrics	
[IFS_NFV_NFVO_16]	NFVO supports provisioning and configuration of network forwarding paths	
[IFS_NFV_NFVO_17]	NFVO supports interaction with WIM for provisioning Transport SDN networks	
[IFS_NFV_VNFM_4]	VNFM supports VNF scaling out/in by adding/removing VNFC instances	
[IFS_NFV_VNFM_6]	VNFM supports VNF scaling out/in request from VNF/EM	
[IFS_NFV_VNFM_7]	VNFM supports receiving VNF indicators from VNF/EM (notifications)	
[IFS_NFV_VNFM_8]	VNFM supports requesting VNF indicators from VNF/EM (query)	
[IFS_NFV_VNFM_11]	VNFM supports receiving virtualised resource performance metrics from VIM	
[IFS_NFV_VNFM_14]	VNFM supports receiving virtualised resource faults/alarms	

[IFS_NFV_VNFM_17]	VNFM supports automatic VNF scaling triggered by VNF indicators	
[IFS_NFV_VNFM_18]	VNFM supports automatic VNF scaling out/in triggered by performance metrics	
[IFS_NFV_VNFM_19]	VNFM supports VNFs with EPA requirements	

A.2 IFS for VIM/NFVI

IFS_ID	Description	Support
[IFS_NFV_VIM_NFVI_1]	VIM exposes OpenStack based NB APIs	
[IFS_NFV_VIM_NFVI_2]	VIM exposes other NB APIs (vCD, ..)	
[IFS_NFV_VIM_NFVI_3]	VIM exposes performance metrics	
[IFS_NFV_VIM_NFVI_4]	VIM exposes alarms	
[IFS_NFV_VIM_NFVI_5]	VIM offers network forwarding path functionality	
[IFS_NFV_VIM_NFVI_6]	NFVI/VIM supports NSH	
[IFS_NFV_VIM_NFVI_7]	NFVI/VIM supports EPA	

A.3 IFS for VNF

IFS_ID	Description	Support
[IFS_NFV_VNF_1]	VNF has its own VNFM	
[IFS_NFV_VNF_2]	VNF can work with generic VNFM	
[IFS_NFV_VNF_3]	VNF can scale out/in by adding/removing VNFC instances	
[IFS_NFV_VNF_4]	VNF supports scale to level	
[IFS_NFV_VNF_5]	VNF/EM can request scaling to VNFM	
[IFS_NFV_VNF_6]	VNF can send VNF Indicators to VNFM (notifications)	
[IFS_NFV_VNF_7]	VNF can send VNF Indicators to VNFM (query response)	
[IFS_NFV_VNF_8]	VNF supports Network Service Headers (NSH) encapsulation	
[IFS_NFV_VNF_9]	VNF requires EPA	

A.4 IFS for VNFM

IFS_ID	Description	Support
[IFS_NFV_VNFM_1]	VNFM supports direct mode (Resource management by VNFM)	

[IFS_NFV_VNFM_2]	VNFM supports in-direct mode (Resource management by NFVO)	
[IFS_NFV_VNFM_3]	VNFM supports multi-site deployments (i.e. two or more geographically distributed sites managed by different VIM instances)	
[IFS_NFV_VNFM_4]	VNFM supports VNF scaling out/in by adding/removing VNFC instances	
[IFS_NFV_VNFM_5]	VNFM supports scale-to-level	
[IFS_NFV_VNFM_6]	VNFM supports VNF scaling out/in request from VNF/EM	
[IFS_NFV_VNFM_7]	VNFM supports receiving VNF indicators from VNF/EM (notifications)	
[IFS_NFV_VNFM_8]	VNFM supports requesting VNF indicators from VNF/EM (query)	
[IFS_NFV_VNFM_9]	VNFM exposes VNF Indicators towards NFVO (notifications)	
[IFS_NFV_VNFM_10]	VNFM exposes VNF Indicators towards NFVO (query response)	
[IFS_NFV_VNFM_11]	VNFM supports receiving virtualised resource performance metrics from VIM	
[IFS_NFV_VNFM_12]	VNFM exposes VNF performance metrics towards NFVO (query response)	
[IFS_NFV_VNFM_13]	VNFM exposes VNF performance metrics towards NFVO (notifications)	
[IFS_NFV_VNFM_14]	VNFM supports receiving virtualised resource faults/alarms from VIM	
[IFS_NFV_VNFM_15]	VNFM exposes VNF alarms towards NFVO (query response)	
[IFS_NFV_VNFM_16]	VNFM exposes VNF alarms towards NFVO (notifications)	
[IFS_NFV_VNFM_17]	VNFM supports automatic VNF scaling triggered by VNF indicators from VNF/EM	
[IFS_NFV_VNFM_18]	VNFM supports automatic scaling out/in triggered by performance metrics from VIM	
[IFS_NFV_VNFM_19]	VNFM supports VNFs with EPA requirements	

A.5 IFS for NS

IFS_ID	Description	Support
[IFS_NFV_NS_1]	NS requires own VNFM(s)	
[IFS_NFV_NS_2]	NS can work with generic VNFM	
[IFS_NFV_NS_3]	VNF(s) in NS can scale out/in by adding/removing VNFC instances	
[IFS_NFV_NS_4]	VNF(s) in the NS can scale to level	
[IFS_NFV_NS_5]	VNF(s) in the NS can request scaling to VNFM	
[IFS_NFV_NS_6]	VNF(s) in NS can send VNF Indicators to VNFM (notifications)	
[IFS_NFV_NS_7]	VNF(s) in NS can send VNF Indicators to VNFM (query response)	
[IFS_NFV_NS_8]	VNF(s) in NS supports Network Service Headers (NSH) encapsulation	

[IFS_NFV_NS_9]	VNF(s) in NS requires EPA	
[IFS_NFV_NS_10]	NS can scale out/in by adding/removing VNF instances	
[IFS_NFV_NS_11]	NS supports scale to level	

History

Document history		
V1.0.0	30/08/2018	Publication