



## **Network Functions Virtualisation (NFV); Testing; Guidelines on Interoperability Testing for MANO**

### *Disclaimer*

---

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.  
It does not necessarily represent the views of the entire ETSI membership.

---

**Reference**

---

DGR/NFV-TST007

---

**Keywords**

---

interoperability, management, MANO, NFV,  
testing**ETSI**

---

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

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

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

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

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

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

---

**Copyright Notification**

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

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2017.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

**3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**oneM2M** logo is protected for the benefit of its Members.

**GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	6
Foreword.....	6
Modal verbs terminology.....	6
1 Scope .....	7
2 References .....	7
2.1 Normative references .....	7
2.2 Informative references.....	7
3 Definitions and abbreviations.....	8
3.1 Definitions.....	8
3.2 Abbreviations .....	8
4 Interoperability Features Statement (IFS) .....	9
4.1 Introduction .....	9
4.2 IFS for VIM.....	9
4.2.1 Software Image Management .....	9
4.2.2 VNF Package Management .....	10
4.2.3 VNF Lifecycle Management.....	10
4.2.4 Fault Management .....	12
4.2.5 Performance Management .....	13
4.2.6 NS Lifecycle Management .....	14
4.3 IFS for VNFM.....	17
4.3.1 Software Image Management .....	17
4.3.2 VNF Package Management .....	17
4.3.3 VNF Lifecycle Management.....	18
4.3.4 Fault Management .....	20
4.3.5 Performance Management .....	20
4.3.6 NS Lifecycle Management .....	21
4.4 IFS for NFVO.....	24
4.4.1 Software Image Management .....	24
4.4.2 VNF Package Management .....	25
4.4.3 VNF Lifecycle Management.....	25
4.4.4 Fault Management .....	27
4.4.5 Performance Management .....	27
4.4.6 NS Lifecycle Management .....	28
4.5 IFS for EM/VNF .....	31
4.5.1 Software Image Management .....	31
4.5.2 VNF Package Management .....	31
4.5.3 VNF Lifecycle Management.....	31
4.5.4 Fault Management .....	31
4.5.5 Performance Management .....	32
4.5.6 NS Lifecycle Management .....	32
5 System Under Test (SUT).....	32
5.1 SUT Configuration 1 .....	32
5.2 SUT Configuration 2 .....	33
5.3 SUT Configuration 3 .....	33
6 Test Suite Structure .....	34
6.1 Introduction .....	34
6.2 Software Image Management Test Cases Overview .....	34
6.3 VNF Package Management Test Cases Overview .....	34
6.4 VNF Lifecycle Management Test Cases Overview .....	35
6.5 Fault Management Test Cases Overview .....	35
6.6 Performance Management Test Cases Overview .....	36
6.7 NS Lifecycle Management Test Cases Overview .....	37

7	Test Descriptions .....	38
7.1	Introduction .....	38
7.2	Software Image Management .....	38
7.2.1	Add Software Image .....	38
7.2.2	Query Software Image .....	38
7.2.2.1	Query Software Image by NFVO .....	38
7.2.2.2	Query Software Image by VNFM .....	39
7.2.3	Update Software Image .....	39
7.2.4	Delete Software Image .....	40
7.3	VNF Package Management .....	40
7.3.1	On-board VNF Package .....	40
7.3.2	Delete VNF Package .....	41
7.3.3	Abort VNF Package Delete Operation .....	41
7.4	VNF Lifecycle Management .....	42
7.4.1	Instantiate VNF with an EM Request .....	42
7.4.2	Query VNF with an EM Request .....	43
7.4.3	Modify VNF Configuration Information with an EM Request .....	44
7.4.4	Start VNF/VNFC with an EM Request .....	44
7.4.5	Stop VNF/VNFC with an EM Request .....	45
7.4.6	VNF Healing with an EM/VNF Request .....	46
7.4.7	VNF Scale Out with an EM/VNF Request .....	47
7.4.8	VNF Scale In with an EM/VNF Request .....	48
7.4.9	Terminate VNF with an EM Request .....	49
7.5	Fault Management .....	50
7.5.1	Virtualised Resource Fault Management .....	50
7.5.1.1	Virtualised Resource Fault Alarm Notification .....	50
7.5.1.2	Virtualised Resource Fault Alarm Clearance Notification .....	51
7.5.2	VNF Fault Management .....	52
7.5.2.1	VNF Fault Alarm Notifications .....	52
7.5.2.2	VNF Fault Alarm Clearance Notifications .....	53
7.6	Performance Management .....	54
7.6.1	Virtualised Resource Performance Management .....	54
7.6.1.1	Virtualised Resource PM Job Creation and Notification Monitoring .....	54
7.6.1.2	Virtualised Resource PM Job Creation and Threshold Monitoring .....	55
7.6.1.3	Virtualised Resource PM Job Deletion .....	56
7.6.1.4	Virtualised Resource PM Threshold Deletion .....	56
7.6.2	VNF Performance Management .....	57
7.6.2.1	VNF PM Job Creation and Notification Monitoring .....	57
7.6.2.2	VNF PM Job Creation and Threshold Monitoring .....	58
7.6.2.3	VNF PM Job Deletion .....	59
7.6.2.4	VNF PM Threshold Deletion .....	60
7.7	NS Lifecycle Management .....	61
7.7.1	NS Instantiation .....	61
7.7.1.1	Standalone NS Instantiation .....	61
7.7.1.2	Nested NS Instantiation .....	62
7.7.2	NS Scaling .....	63
7.7.2.1	NS Scale Out .....	63
7.7.2.1.1	NS Scale out with an Operator Action .....	63
7.7.2.1.2	NS Scale out with a VNF Indicator .....	64
7.7.2.1.3	NS Scale out with a VIM KPI .....	65
7.7.2.2	NS Scale In .....	66
7.7.2.2.1	NS Scale In with an Operator Action .....	66
7.7.2.2.2	NS Scale in with a VNF Indicator .....	67
7.7.2.2.3	NS Scale in with a VIM KPI .....	68
7.7.2.3	NS VNF Scale Out .....	69
7.7.2.3.1	NS VNF Scale Out with an Operator Action .....	69
7.7.2.3.2	NS VNF Scale Out with a VNF Indicator .....	70
7.7.2.3.3	NS VNF Scale Out with a VIM KPI .....	71
7.7.2.4	NS VNF Scale In .....	72
7.7.2.4.1	NS VNF Scale In with an Operator Action .....	72
7.7.2.4.2	NS VNF Scale In with a VNF Indicator .....	73
7.7.2.4.3	NS VNF Scale In with a VIM KPI .....	74

7.7.3	NS Update.....	75
7.7.3.1	Start VNF Instance.....	75
7.7.3.2	Stop VNF Instance.....	76
7.7.3.3	Instantiate VNF and Add Instance to NS Instance.....	77
7.7.3.4	Remove VNF Instances from a NS Instance.....	78
7.7.3.5	Add Shared VNF Instances to NS Instance.....	79
7.7.3.6	Remove Shared VNF Instances from NS Instance.....	80
7.7.3.7	Change VNF Deployment Flavour.....	81
7.7.4	NS Healing.....	82
7.7.4.1	Partial NS Healing with an Operator Action.....	82
7.7.4.2	Complete NS Healing with an Operator Action.....	83
7.7.5	NS Termination.....	84
7.7.5.1	Standalone NS Termination.....	84
7.7.5.2	Nested NS Termination.....	85
<b>Annex A:</b>	<b>Technical Report Card Sample.....</b>	<b>86</b>
<b>Annex B:</b>	<b>Document Usage Process Diagram.....</b>	<b>87</b>
<b>Annex C:</b>	<b>Authors &amp; Contributors.....</b>	<b>88</b>
History.....		89

---

## Intellectual Property Rights

### Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

### Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

---

## Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

---

## Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

# 1 Scope

The present document describes a set of informative interoperability test guidelines for NFV capabilities that require interactions between the components implementing NFV functionality, namely, the NFVO, VNFM, EM-VNF and VIM-NFVI (Functions under Test).

The guidelines in the present document follow the interoperability testing methodology described in ETSI GS NFV-TST 002 [i.1] and are implementation agnostic. The capabilities in scope are enabled by the interfaces on the main NFV reference points between the Functions Under Test:

- Or-Vi;
- Or-Vnfm;
- Vi-Vnfm; and
- Ve-Vnfm.

---

## 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.

- [i.1] ETSI GS NFV-TST 002 (V1.1.1): "Network Functions Virtualisation (NFV); Testing Methodology; Report on NFV Interoperability Testing Methodology".
- [i.2] ETSI GS NFV 003 (V1.2.1): "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".
- [i.3] ETSI GS NFV-IFA 005 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".
- [i.4] ETSI GS NFV-IFA 006 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".
- [i.5] ETSI GS NFV-IFA 007 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Or-Vnfm reference point - Interface and Information Model Specification".
- [i.6] ETSI GS NFV-IFA 008 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Ve-Vnfm reference point - Interface and Information Model Specification".

- [i.7] ETSI GS NFV-IFA 010 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Functional requirements specification".
- [i.8] ETSI GS NFV-IFA 011 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; VNF Packaging Specification".
- [i.9] ETSI GS NFV-IFA 013 (V2.3.1): "Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Os-Ma-Nfvo reference point - Interface and Information Model Specification".

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS NFV 003 [i.2] apply.

NOTE: A term defined in the present document takes precedence over the definition of the same term, if any, in ETSI GS NFV 003 [i.2].

### 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [i.2] and the following apply:

NOTE: An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in ETSI GS NFV 003 [i.2].

CRM	Compute Resource Management
DF	Deployment Flavour
EM	Element Manager
FM	Fault Management
IFS	Interoperability Features Statement
KPI	Key Performance Indicators
MANO	Management and Orchestration
NFV	Network Function Virtualisation
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NIC	Network Interface Controller
NS	Network Service
NSD	NS Descriptor
NRM	Network Resource Management
PM	Performance Management
SRM	Storage Resource Management
SUT	System Under Test
VIM	Virtual Infrastructure Manager
VL	Virtual Link
VLD	VL Descriptor
VM	Virtual Machine
VNF	Virtual Network Functions
VNFC	VNF Component
VNFD	VNF Descriptor
VNFFG	VNF Forwarding Graph
VNFFGD	VNFFG Descriptor
VNFM	VNF Manager
VR	Virtual Resource



## 4 Interoperability Features Statement (IFS)

### 4.1 Introduction

This clause compiles the Interoperability Feature Statement (IFS) for VIM, VNFM, EM/VNF and NFVO according to clause 4.3.3 of ETSI GS NFV-TST 002 [i.1] and following the example in ETSI GS NFV-TST 002 [i.1], annex A.

The interoperability feature statement identifier starts with a reference to the relevant functional block that produces or consumes the interface:

- IDs starting with "VIM\_" designate features on interfaces produced or consumed by the VIM
- IDs starting with "VNFM\_" designate features on interfaces produced or consumed by the VNFM
- IDs starting with "NFVO\_" designate features on interfaces produced or consumed by the NFVO
- IDs starting with "EM\_" designate features on interfaces produced or consumed by the EM/VNF

The purpose of the Interoperable Features Statement (IFS) is to identify those standardized functions which a FUT supports, including those which are optional and those which are conditional on the support of other functions. Supported interoperability feature statements should be indicated by Y in the Support column (or N if not supported). The Details column can be used to add further support details when necessary.

### 4.2 IFS for VIM

#### 4.2.1 Software Image Management

**Table 4.2.1-1**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Software Image Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_SWIM_QUERY_IM_BY_NFVO	VIM supports software image information queries by the NFVO	M		
VIM_SWIM_ADD_IM	VIM supports "add image" operations by the NFVO	M		
VIM_SWIM_UPDATE_IM	VIM supports "update image" operations by the NFVO	M		
VIM_SWIM_DELETE_IM	VIM supports "delete image" operations by the NFVO	M		

**Table 4.2.1-2**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Software Image Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_SWIM_QUERY_IM_BY_VNFM	VIM supports software image information queries by the VNFM	M		

## 4.2.2 VNF Package Management

No IFS needed for VNF Package Management on the VIM.

## 4.2.3 VNF Lifecycle Management

**Table 4.2.3-1**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_CRM_ALLOCATE_BY_NFVO	VIM supports "allocate compute resource" operation requests from the NFVO	M		
VIM_CRM_INFO_TO_NFVO	VIM can send compute resource information to the NFVO	M		
VIM_CRM_OPERATE_BY_NFVO	VIM supports "operate compute resource" operation requests from the NFVO	M		
VIM_CRM_TERMINATE_BY_NFVO	VIM supports "terminate compute resource" operation requests from the NFVO	M		

**Table 4.2.3-2**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_CRM_ALLOCATE_BY_VNFM	VIM supports "allocate compute resource" operation requests from the VNFM	M		
VIM_CRM_INFO_TO_VNFM	VIM can send compute resource information to the VNFM	M		
VIM_CRM_OPERATE_BY_VNFM	VIM supports "operate compute resource" operation requests from the VNFM	M		
VIM_CRM_TERMINATE_BY_VNFM	VIM supports "terminate compute resource" operation requests from the VNFM	M		

Table 4.2.3-3

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_NRM_ALLOCATE_BY_NFVO	VIM supports "allocate network resource" operation requests from the NFVO	M		
VIM_NRM_INFO_TO_NFVO	VIM can send network resource information to the NFVO	M		
VIM_NRM_TERMINATE_BY_NFVO	VIM supports "terminate network resource" operation requests from the NFVO	M		

Table 4.2.3-4

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Network Resource Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_NRM_ALLOCATE_BY_VNFM	VIM supports "allocate network resource" operation requests from the VNFM	M		
VIM_NRM_INFO_TO_VNFM	VIM can send network resource information to the VNFM	M		
VIM_NRM_TERMINATE_BY_VNFM	VIM supports "terminate network resource" operation requests from the VNFM	M		

Table 4.2.3-5

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_SRM_ALLOCATE_BY_NFVO	VIM supports "allocate storage resource" operation requests from the NFVO	M		
VIM_SRM_INFO_TO_NFVO	VIM can send storage resource information to the NFVO	M		
VIM_SRM_TERMINATE_BY_NFVO	VIM supports "terminate storage resource" operation requests from the NFVO	M		

Table 4.2.3-6

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_SRM_ALLOCATE_BY_VNFM	VIM supports "allocate storage resource" operation requests from the VNFM	M		
VIM_SRM_INFO_TO_VNFM	VIM can send storage resource information to the VNFM	M		
VIM_SRM_TERMINATE_BY_VNFM	VIM supports "terminate storage resource" operation requests from the VNFM	M		

## 4.2.4 Fault Management

Table 4.2.4-1

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Resources Fault Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_FM_SUBSCRIBE_BY_NFVO	VIM supports alarm notifications subscriptions by the NFVO	M		
VIM_FM_NOTIFY_BY_NFVO	VIM can generate virtualised resources fault alarm notifications to the NFVO	M		
VIM_FM_QUERY_BY_NFVO	VIM supports virtualised resource fault alarm list queries by the NFVO	M		

Table 4.2.4-2

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Resources Fault Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_FM_SUBSCRIBE_BY_VNFM	VIM supports alarm notifications subscriptions by the VNFM	M		
VIM_FM_NOTIFY_BY_VNFM	VIM can generate virtualised resources fault alarm notifications to the VNFM	M		
VIM_FM_QUERY_BY_VNFM	VIM supports virtualised resources fault alarm list queries by the VNFM	M		

## 4.2.5 Performance Management

Table 4.2.5-1

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Resources Performance Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_PM_PMJOB_CREATE_BY_NFVO	VIM supports VR PM jobs creation by the NFVO	M		
VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO	VIM supports VR PM job subscriptions from the NFVO	M		
VIM_PM_PMJOB_NOTIFY_BY_NFVO	VIM can generate VR PM notifications to the NFVO	M		
VIM_PM_PMJOB_QUERY_BY_NFVO	VIM supports VR PM job queries by the NFVO	M		
VIM_PM_PMJOB_DELETE_BY_NFVO	VIM supports VR PM jobs deletion by the NFVO	M		
VIM_PM_PMTTH_CREATE_BY_NFVO	VIM supports VR PM thresholds creation by the NFVO	M		
VIM_PM_PMTTH_QUERY_BY_NFVO	VIM supports VR PM threshold queries by the NFVO	M		
VIM_PM_PMTTH_DELETE_BY_NFVO	VIM supports VR PM thresholds deletion by the NFVO	M		

Table 4.2.5-2

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	Performance Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Resources Performance Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_PM_PMJOB_CREATE_BY_VNFM	VIM supports VR PM jobs creation by the VNFM	M		
VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM	VIM supports VR PM job subscriptions from the VNFM	M		
VIM_PM_PMJOB_NOTIFY_BY_VNFM	VIM can generate VR PM notifications to the VNFM	M		
VIM_PM_PMJOB_QUERY_BY_VNFM	VIM supports VR PM job queries by the VNFM	M		
VIM_PM_PMJOB_DELETE_BY_VNFM	VIM supports VR PM jobs deletion by the VNFM	M		
VIM_PM_PMTTH_CREATE_BY_VNFM	VIM supports VR PM thresholds creation by the VNFM	M		
VIM_PM_PMTTH_QUERY_BY_VNFM	VIM supports VR PM threshold queries by the VNFM	M		
VIM_PM_PMTTH_DELETE_BY_VNFM	VIM supports VR PM thresholds deletion by the VNFM	M		

## 4.2.6 NS Lifecycle Management

**Table 4.2.6-1**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_SWIM_QUERY_IM_BY_NFVO	VIM supports software image information queries by the NFVO	M		

**Table 4.2.6-2**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_SWIM_QUERY_IM_BY_VNFM	VIM supports software image information queries by the VNFM	M		

**Table 4.2.6-3**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_CRM_ALLOCATE_BY_NFVO	VIM supports "allocate compute resource" operation requests from the NFVO	M		
VIM_CRM_TERMINATE_BY_NFVO	VIM supports "terminate compute resource" operation requests from the NFVO	M		
VIM_CRM_INFO_TO_NFVO	VIM can send compute resource information to the NFVO	M		
VIM_CRM_OPERATE_BY_NFVO	VIM supports "operate compute resource" operation requests from the NFVO	M		

**Table 4.2.6-4**

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			

Interoperability Features				
Id	Feature	Status	Support	Details
VIM_NRM_ALLOCATE_BY_NFVO	VIM supports "allocate network resource" operation requests from the NFVO	M		
VIM_NRM_TERMINATE_BY_NFVO	VIM supports "terminate network resource" operation requests from the NFVO	M		
VIM_NRM_INFO_TO_NFVO	VIM can send network resource information to the NFVO	M		

Table 4.2.6-5

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
Interoperability Features				
Id	Feature	Status	Support	Details
VIM_SRM_ALLOCATE_BY_NFVO	VIM supports "allocate storage resource" operation requests from the NFVO	M		
VIM_SRM_TERMINATE_BY_NFVO	VIM supports "terminate storage resource" operation requests from the NFVO	M		
VIM_SRM_INFO_TO_NFVO	VIM can send storage resource information to the NFVO	M		

Table 4.2.6-6

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
Interoperability Features				
Id	Feature	Status	Support	Details
VIM_CRM_ALLOCATE_BY_VNFM	VIM supports "allocate compute resource" operation requests from the VNFM	M		
VIM_CRM_TERMINATE_BY_VNFM	VIM supports "terminate compute resource" operation requests from the VNFM	M		
VIM_CRM_INFO_TO_VNFM	VIM can send compute resource information to the VNFM	M		
VIM_CRM_OPERATE_BY_VNFM	VIM supports "operate compute resource" operation requests from the VNFM	M		

Table 4.2.6-7

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			

Interoperability Features				
Id	Feature	Status	Support	Details
VIM_NRM_ALLOCATE_BY_VNFM	VIM supports "allocate network resource" operation requests from the VNFM	M		
VIM_NRM_TERMINATE_BY_VNFM	VIM supports "terminate network resource" operation requests from the VNFM	M		
VIM_NRM_INFO_TO_VNFM	VIM can send network resource information to the VNFM	M		

Table 4.2.6-8

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
Interoperability Features				
Id	Feature	Status	Support	Details
VIM_SRM_ALLOCATE_BY_VNFM	VIM supports "allocate storage resource" operation requests from the VNFM	M		
VIM_SRM_TERMINATE_BY_VNFM	VIM supports "terminate storage resource" operation requests from the VNFM	M		
VIM_SRM_INFO_TO_VNFM	VIM can send storage resource information to the VNFM	M		

Table 4.2.6-9

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Resources Performance Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
Interoperability Features				
Id	Feature	Status	Support	Details
VIM_PM_PMJOB_CREATE_BY_NFVO	VIM supports VR PM jobs creation by the NFVO	M		
VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO	VIM supports VR PM job subscriptions from the NFVO	M		
VIM_PM_PMJOB_NOTIFY_BY_NFVO	VIM can generate VR PM notifications to the NFVO	M		
VIM_PM_PMTN_CREATE_BY_NFVO	VIM supports VR PM thresholds creation by the NFVO	M		



Table 4.2.6-10

<b>Functional Block</b>	VIM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Resources Performance Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VIM_PM_PMJOB_CREATE_BY_VNFM	VIM supports VR PM jobs creation by the VNFM	M		
VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM	VIM supports VR PM job subscriptions from the VNFM	M		
VIM_PM_PMJOB_NOTIFY_BY_VNFM	VIM can generate VR PM notifications to the VNFM	M		
VIM_PM_PMTHTH_CREATE_BY_VNFM	VIM supports VR PM thresholds creation by the VNFM	M		

## 4.3 IFS for VNFM

### 4.3.1 Software Image Management

Table 4.3.1-1

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	Software Image Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_SWIM_QUERY_IM	VNFM can query software image information on the VIM	M		

### 4.3.2 VNF Package Management

No IFS needed for VNF Package Management on the VNFM.

### 4.3.3 VNF Lifecycle Management

**Table 4.3.3-1**

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_CRM_QUERY	VNFM can query compute resource information from the VIM	M		
VNFM_CRM_ALLOCATE	VNFM can generate "allocate compute resource" operation requests to the VIM	M		
VNFM_CRM_OPERATE	VNFM can generate "operate compute resource" operation requests to the VIM	M		
VNFM_CRM_TERMINATE	VNFM can generate "terminate compute resource" operation requests to the VIM	M		

**Table 4.3.3-2**

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_NRM_QUERY	VNFM can query network resource information from the VIM	M		
VNFM_NRM_ALLOCATE	VNFM can generate "allocate network resource" operation requests to the VIM	M		
VNFM_NRM_TERMINATE	VNFM can generate "terminate network resource" operation requests to the VIM	M		

**Table 4.3.3-3**

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_SRM_QUERY	VNFM can query storage resource information from the VIM	M		
VNFM_SRM_ALLOCATE	VNFM can generate "allocate storage resource" operation requests to the VIM	M		
VNFM_SRM_TERMINATE	VNFM can generate "terminate storage resource" operation requests to the VIM	M		

Table 4.3.3-4

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_VNFLCM_QUERY	VNFM supports VNF information queries by the NFVO	M		

Table 4.3.3-5

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Ve-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 008 [i.6]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_VNFLCM_VNF_INSTANTIATE	VNFM supports "instantiate VNF" requests from the EM	M		
VNFM_VNFLCM_VNF_QUERY	VNFM supports "query VNF" requests from the EM	M		
VNFM_VNFLCM_VNF_MODIFY_INFO	VNFM supports "modify VNF information" requests from the EM	M		
VNFM_VNFLCM_VNF_OPERATE	VNFM supports "operate VNF" requests from the EM	M		
VNFM_VNFLCM_VNF_HEAL	VNFM supports "heal VNF" requests from the EM/VNF	M		
VNFM_VNFLCM_VNF_SCALE_OUT	VNFM supports "scale out by adding VNFC instances" requests from the EM/VNF	M		
VNFM_VNFLCM_VNF_SCALE_IN	VNFM supports "scale in by removing VNFC instances" requests from the EM/VNF	M		
VNFM_VNFLCM_VNF_TERMINATE	VNFM supports "terminate VNF" requests from the EM	M		

Table 4.3.3-6

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Operation Granting			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_VNFLCM_GRANTING	VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO	O		

## 4.3.4 Fault Management

**Table 4.3.4-1**

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Resources Fault Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_FM_VR_SUBSCRIBE	VNFM can subscribe to alarm notifications on the VIM	M		
VNFM_FM_VR_NOTIFY	VNFM can process virtualised resource fault alarm notifications from by the VIM	M		
VNFM_FM_VR_GET_ALARM	VNFM can get the list of virtualised resource fault alarms from the VIM	M		

**Table 4.3.4-2**

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Fault Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_FM_VNF_SUBSCRIBE	VNFM supports alarm notifications subscriptions from by the NFVO	M		
VNFM_FM_VNF_NOTIFY	VNFM can generate VNF fault alarm notifications to the NFVO	M		
VNFM_FM_VNF_QUERY	VNFM supports VNF fault alarm list queries by the NFVO	M		

## 4.3.5 Performance Management

**Table 4.3.5-1**

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	Performance Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Resources Performance Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			

Interoperability Features				
Id	Feature	Status	Support	Details
VNFM_PM_VR_PMJOB_CREATE	VNFM can create VR PM jobs on the VIM	M		
VNFM_PM_VR_PMJOB_SUBSCRIBE	VNFM can subscribe to VR PM jobs on the VIM	M		
VNFM_PM_VR_PMJOB_NOTIFY	VNFM supports VR PM job notifications from the VIM	M		
VNFM_PM_VR_PMJOB_QUERY	VNFM can query VR PM jobs from the VIM	M		
VNFM_PM_VR_PMJOB_DELETE	VNFM can delete VR PM jobs from the VIM	M		
VNFM_PM_VR_PMTH_CREATE	VNFM can create VR PM thresholds on the VIM	M		
VNFM_PM_VR_PMTH_QUERY	VNFM can query VR PM thresholds from the VIM	M		
VNFM_PM_VR_PMTH_DELETE	VNFM can delete VR PM thresholds from the VIM	M		

Table 4.3.5-2

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	Performance Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Performance Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
Interoperability Features				
Id	Feature	Status	Support	Details
VNFM_PM_VNF_PMJOB_CREATE	VNFM supports VNF PM jobs creation by the NFVO	M		
VNFM_PM_VNF_PMJOB_SUBSCRIBE	VNFM supports VNF PM job subscriptions from the NFVO	M		
VNFM_PM_VNF_PMJOB_NOTIFY	VNFM can generate VNF PM notifications to the NFVO	M		
VNFM_PM_VNF_PMJOB_QUERY	VNFM supports VNF PM job queries by the NFVO	M		
VNFM_PM_VNF_PMJOB_DELETE	VNFM supports VNF PM jobs deletion by the NFVO	M		
VNFM_PM_VNF_PMTH_CREATE	VNFM supports VNF PM thresholds creation by the NFVO	M		
VNFM_PM_VNF_PMTH_QUERY	VNFM supports VNF PM threshold queries by the NFVO	M		
VNFM_PM_VNF_PMTH_DELETE	VNFM supports VNF PM thresholds deletion by the NFVO	M		

## 4.3.6 NS Lifecycle Management

Table 4.3.6-1

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
Interoperability Features				
Id	Feature	Status	Support	Details
VNFM_SWIM_QUERY_IM	VNFM can query software image information on the VIM	M		

Table 4.3.6-2

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_CRM_QUERY	VNFM can query compute resource information from the VIM	M		
VNFM_CRM_ALLOCATE	VNFM can generate "allocate compute resource" operation requests to the VIM	M		
VNFM_CRM_OPERATE	VNFM can generate "operate compute resource" operation requests to the VIM	M		
VNFM_CRM_TERMINATE	VNFM can generate "terminate compute resource" operation requests to the VIM	M		

Table 4.3.6-3

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_NRM_QUERY	VNFM can query network resource information from the VIM	M		
VNFM_NRM_ALLOCATE	VNFM can generate "allocate network resource" operation requests to the VIM	M		
VNFM_NRM_TERMINATE	VNFM can generate "terminate network resource" operation requests to the VIM	M		

Table 4.3.6-4

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_SRM_QUERY	VNFM can query storage resource information from the VIM	M		
VNFM_SRM_ALLOCATE	VNFM can generate "allocate storage resource" operation requests to the VIM	M		
VNFM_SRM_TERMINATE	VNFM can generate "terminate storage resource" operation requests to the VIM	M		

Table 4.3.6-5

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Vi-Vnfm			
<b>Observed Interface</b>	Virtualised Resources Performance Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_PM_VR_PMJOB_CREATE	VNFM can create VR PM jobs on the VIM	M		
VNFM_PM_VR_PMJOB_SUBSCRIBE	VNFM can subscribe to VR PM jobs on the VIM	M		
VNFM_PM_VR_PMJOB_NOTIFY	VNFM supports VR PM job notifications from the VIM	M		

Table 4.3.6-6

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_VNFLCM_CREATE_VNFID	VNFM supports "create VNF identifier" operation requests from the NFVO	M		
VNFM_VNFLCM_INSTANTIATE	VNFM supports "instantiate VNF" operation requests from the NFVO	M		
VNFM_VNFLCM_DELETE_VNFID	VNFM supports "delete VNF identifier" operation requests from the NFVO	M		
VNFM_VNFLCM_TERMINATE	VNFM supports "terminate VNF" operation requests from the NFVO	M		
VNFM_VNFLCM_QUERY	VNFM supports VNF information queries by the NFVO	M		
VNFM_VNFLCM_HEAL	VNFM supports VNF healing requests from the NFVO	M		
VNFM_VNFLCM_OPERATE	VNFM supports "operate VNF" operation requests from the NFVO	M		
VNFM_VNFLCM_NS_SCALE_OUT	VNFM supports "scale out by adding VNF instances" requests from the NFVO	M		
VNFM_VNFLCM_NS_SCALE_IN	VNFM supports "scale in by removing VNF instances" requests from the NFVO	M		
VNFM_VNFLCM_VNF_SCALE_OUT	VNFM supports "scale out by adding VNFC instances" requests from the NFVO	M		
VNFM_VNFLCM_VNF_SCALE_IN	VNFM supports "scale in by removing VNFC instances" requests from the NFVO	M		

Table 4.3.6-7

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	Indicator			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_NSVMFINDI_SUBSCRIBE	VNFM supports VNF indicator subscriptions from the NFVO	M		
VNFM_NSVMFINDI_NOTIFY	VNFM can generate VNF indicator notifications to the NFVO	M		

Table 4.3.6-8

<b>Functional Block</b>	VNFM			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Ve-Vnfm			
<b>Observed Interface</b>	Indicator			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 008 [i.6]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_VNFINDI_SUBSCRIBE	VNFM can subscribe to VNF indicators on the EM/VNF	M		
VNFM_VNFINDI_NOTIFY	VNFM supports VNF indicator notifications from the EM/VNF	M		

## 4.4 IFS for NFVO

### 4.4.1 Software Image Management

Table 4.4.1-1

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	Software Image Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_SWIM_ADD_IM	NFVO can add software images on the VIM	M		
NFVO_SWIM_QUERY_IM	NFVO can query software image information on the VIM	M		
NFVO_SWIM_UPDATE_IM	NFVO can update software image information on the VIM	M		
NFVO_SWIM_DELETE_IM	NFVO can delete software images from the VIM	M		



## 4.4.2 VNF Package Management

**Table 4.4.2-1**

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Package Management			
<b>Observed Reference Point</b>	Os-Ma-Nfvo			
<b>Observed Interface</b>	VNF Package Management			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 013 [i.9]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_VNFPM_VALIDATE	NFVO can check the integrity and validity of VNF Packages	M		
NFVO_VNFPM_ACCESS	NFVO can access the VNF catalogue where the VNF Packages information is stored	M		
NFVO_VNFPM_QUERY	NFVO supports on-boarded VNF Packages queries	M		
NFVO_VNFPM_ABORT_DELETE	NFVO supports the delete VNF Package operation abortion	M		

## 4.4.3 VNF Lifecycle Management

**Table 4.4.3-1**

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_SWIM_QUERY_IM	NFVO can query software image information on the VIM	M		

**Table 4.4.3-2**

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_CRM_QUERY	NFVO can query compute resource information from the VIM	M		
NFVO_CRM_ALLOCATE	NFVO can generate "allocate compute resource" operation requests to the VIM	M		
NFVO_CRM_OPERATE	NFVO can generate "operate compute resource" operation requests to the VIM	M		
NFVO_CRM_TERMINATE	NFVO can generate "terminate compute resource" operation requests to the VIM	M		

Table 4.4.3-3

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_NRM_QUERY	NFVO can query network resource information from the VIM	M		
NFVO_NRM_ALLOCATE	NFVO can generate "allocate network resource" operation requests to the VIM	M		
NFVO_NRM_TERMINATE	NFVO can generate "terminate network resource" operation requests to the VIM	M		

Table 4.4.3-4

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_SRM_QUERY	NFVO can query storage resource information from the VIM	M		
NFVO_SRM_ALLOCATE	NFVO can generate "allocate storage resource" operation requests to the VIM	M		
NFVO_SRM_TERMINATE	NFVO can generate "terminate storage resource" operation requests to the VIM	M		

Table 4.4.3-5

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_VNFLCM_QUERY	NFVO can query VNF information from the VNFM	M		

Table 4.4.3-6

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Operation Granting			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_VNFLCM_GRANTING	NFVO supports "Grant VNF Lifecycle	O		

	Operation" requests from the VNFM			
--	-----------------------------------	--	--	--

#### 4.4.4 Fault Management

**Table 4.4.4-1**

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Resource Fault Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_FM_VR_SUBSCRIBE	NFVO can subscribe to virtualised resource fault alarms on the VIM	M		
NFVO_FM_VR_NOTIFY	NFVO can process virtualised resource fault alarm notifications from the VIM	M		
NFVO_FM_VR_GET_ALARMS	NFVO can get the list of virtualised resource fault alarms from the VIM	M		

**Table 4.4.4-2**

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	Fault Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Fault Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_FM_VNF_SUBSCRIBE	NFVO can subscribe to alarm notifications on the VNFM	M		
NFVO_FM_VNF_NOTIFY	NFVO can process VNF fault alarm notifications from by the VNFM	M		
NFVO_FM_VNF_GET_ALARMS	NFVO can get the list of VNF fault alarms from the VNFM	M		

#### 4.4.5 Performance Management

**Table 4.4.5-1**

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	Performance Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Resource Performance Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			

Interoperability Features				
Id	Feature	Status	Support	Details
NFVO_PM_VR_PMJOB_CREATE	NFVO can create VR PM jobs on the VIM	M		
NFVO_PM_VR_PMJOB_SUBSCRIBE	NFVO can subscribe to VR PM jobs on the VIM	M		
NFVO_PM_VR_PMJOB_NOTIFY	NFVO supports VR PM notifications from the VIM	M		
NFVO_PM_VR_PMJOB_QUERY	NFVO can query VR PM jobs from the VIM	M		
NFVO_PM_VR_PMJOB_DELETE	NFVO can delete VR PM jobs from the VIM	M		
NFVO_PM_VR_PMTN_CREATE	NFVO can create VR PM thresholds on the VIM	M		
NFVO_PM_VR_PMTN_QUERY	NFVO can query VR PM thresholds from the VIM	M		
NFVO_PM_VR_PMTN_DELETE	NFVO can delete VR PM thresholds from the VIM	M		

Table 4.4.5-2

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	Performance Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Performance Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
Interoperability Features				
Id	Feature	Status	Support	Details
NFVO_PM_VNF_PMJOB_CREATE	NFVO can create VNF PM jobs on the VNFM	M		
NFVO_PM_VNF_PMJOB_SUBSCRIBE	NFVO can subscribe to VNF PM jobs on the VNFM	M		
NFVO_PM_VNF_PMJOB_NOTIFY	NFVO supports VNF PM notifications from the VNFM	M		
NFVO_PM_VNF_PMJOB_QUERY	NFVO can query VNF PM jobs from the VNFM	M		
NFVO_PM_VNF_PMJOB_DELETE	NFVO can create VNF PM jobs on the VNFM	M		
NFVO_PM_VNF_PMTN_CREATE	NFVO can create VNF PM thresholds on the VNFM	M		
NFVO_PM_VNF_PMTN_QUERY	NFVO can query VNF PM thresholds from the VNFM	M		
NFVO_PM_VNF_PMTN_DELETE	NFVO can delete VNF PM thresholds from the VNFM	M		

## 4.4.6 NS Lifecycle Management

Table 4.4.6-1

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Software Image Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
Interoperability Features				
Id	Feature	Status	Support	Details
NFVO_SWIM_QUERY_IM	NFVO can query software image information on the VIM	M		

Table 4.4.6-2

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Compute Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_CRM_QUERY	NFVO can query compute resource information from the VIM	M		
NFVO_CRM_ALLOCATE	NFVO can generate "allocate compute resource" operation requests to the VIM	M		
NFVO_CRM_OPERATE	NFVO can generate "operate compute resource" operation requests to the VIM	M		
NFVO_CRM_TERMINATE	NFVO can generate "terminate compute resource" operation requests to the VIM	M		

Table 4.4.6-3

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Network Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_NRM_QUERY	NFVO can query network resource information from the VIM	M		
NFVO_NRM_ALLOCATE	NFVO can generate "allocate network resource" operation requests to the VIM	M		
NFVO_NRM_TERMINATE	NFVO can generate "terminate network resource" operation requests to the VIM	M		

Table 4.4.6-4

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Storage Resource Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_SRM_QUERY	NFVO can query storage resource information from the VIM	M		
NFVO_SRM_ALLOCATE	NFVO can generate "allocate storage resource" operation requests to the VIM	M		
NFVO_SRM_TERMINATE	NFVO can generate "terminate storage resource" operation requests to the VIM	M		

Table 4.4.6-5

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vi			
<b>Observed Interface</b>	Virtualised Resource Performance Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_PM_VR_PMJOB_CREATE	NFVO can create VR PM jobs on the VIM	M		
NFVO_PM_VR_PMJOB_SUBSCRIBE	NFVO can subscribe to VR PM jobs on the VIM	M		
NFVO_PM_VR_PMJOB_NOTIFY	NFVO supports VR PM notifications from the VIM	M		

Table 4.4.6-6

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
NFVO_VNFLCM_CREATE_VNFID	NFVO can generate "create VNF identifier" operation requests to the VNFM	M		
NFVO_VNFLCM_INSTANTIATE	NFVO can generate "instantiate VNF" operation requests to the VNFM	M		
NFVO_VNFLCM_DELETE_VNFID	NFVO can generate "delete VNF identifier" operation requests to the VNFM	M		
NFVO_VNFLCM_TERMINATE	NFVO can generate "terminate VNF" operation requests to the VNFM	M		
NFVO_VNFLCM_QUERY	NFVO can query VNF information from the VNFM	M		
NFVO_VNFLCM_HEAL	NFVO can generate VNF healing requests to the VNFM	M		
NFVO_VNFLCM_OPERATE	NFVO can generate "operate VNF" operation requests to the VNFM	M		
NFVO_VNFLCM_NS_SCALE_OUT	NFVO can generate "scale out by adding VNF instances" requests to the VNFM	M		
NFVO_VNFLCM_NS_SCALE_IN	NFVO can generate "scale in by removing VNF instances" requests to the VNFM	M		
NFVO_VNFLCM_VNF_SCALE_OUT	NFVO can generate "scale out by adding VNFC instances" to the VNFM	M		
NFVO_VNFLCM_VNF_SCALE_IN	NFVO can generate "scale in by removing VNFC instances" to the VNFM	M		

Table 4.4.6-7

<b>Functional Block</b>	NFVO			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Or-Vnfm			
<b>Observed Interface</b>	Indicator			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
VNFM_NSVNFINDI_SUBSCRIBE	NFVO can subscribe to VNF indicators on the VNFM	M		
VNFM_NSVNFINDI_NOTIFY	NFVO supports VNF indicator notifications from the VNFM	M		

## 4.5 IFS for EM/VNF

### 4.5.1 Software Image Management

No IFS needed for Software Image Management on the EM/VNF.

### 4.5.2 VNF Package Management

No IFS needed for VNF Package Management on the EM/VNF.

### 4.5.3 VNF Lifecycle Management

Table 4.5.3-1

<b>Functional Block</b>	EM/VNF			
<b>Functional Area</b>	VNF Lifecycle Management			
<b>Observed Reference Point</b>	Ve-Vnfm			
<b>Observed Interface</b>	VNF Lifecycle Management			
<b>Producer/Consumer</b>	Consumer			
<b>References</b>	ETSI GS NFV-IFA 008 [i.6]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
EM_VNFLCM_VNF_INSTANTIATE	EM can generate "Instantiate VNF" requests to the VNFM	M		
EM_VNFLCM_VNF_QUERY	EM can generate "query VNF" requests to the VNFM	M		
EM_VNFLCM_VNF_MODIFY_INFO	EM can generate "modify VNF information" requests to the VNFM	M		
EM_VNFLCM_VNF_OPERATE	EM can generate "operate VNF" requests to the VNFM	M		
EM_VNFLCM_VNF_SCALE_OUT	EM/VNF can generate "scale out by adding VNFC instances" requests to the VNFM	M		
EM_VNFLCM_VNF_SCALE_IN	EM/VNF can generate "scale in by removing VNFC instances" requests to the VNFM	M		
EM_VNFLCM_VNF_HEAL	EM/VNF can generate "heal VNF" requests to the VNFM	M		
EM_VNFLCM_VNF_TERMINATE	EM can generate "terminate VNF" requests to the VNFM	M		

### 4.5.4 Fault Management

No IFS needed for Fault Management on the EM/VNF.

## 4.5.5 Performance Management

No IFS needed for Performance Management on the EM/VNF.

## 4.5.6 NS Lifecycle Management

Table 4.5.6-1

<b>Functional Block</b>	EM/VNF			
<b>Functional Area</b>	NS Lifecycle Management			
<b>Observed Reference Point</b>	Ve-Vnfm			
<b>Observed Interface</b>	Indicator			
<b>Producer/Consumer</b>	Producer			
<b>References</b>	ETSI GS NFV-IFA 008 [i.6]			
<b>Interoperability Features</b>				
<b>Id</b>	<b>Feature</b>	<b>Status</b>	<b>Support</b>	<b>Details</b>
EM_VNFINDI_SUBSCRIBE	EM/VNF supports VNF indicator subscriptions from the VNFM	M		
EM_VNFINDI_NOTIFY	EM/VNF can generate VNF indicator notifications to the VNFM	M		

# 5 System Under Test (SUT)

## 5.1 SUT Configuration 1

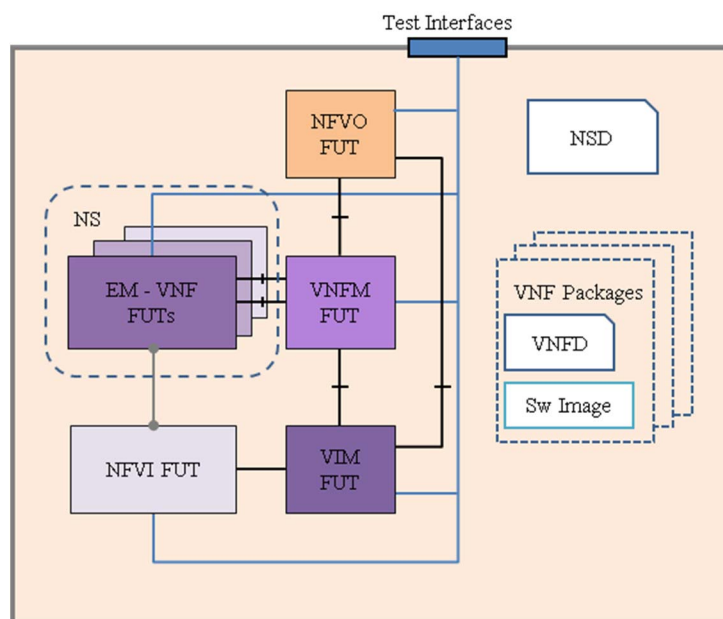


Figure 5.1-1: SUT Configuration 1



## 5.2 SUT Configuration 2

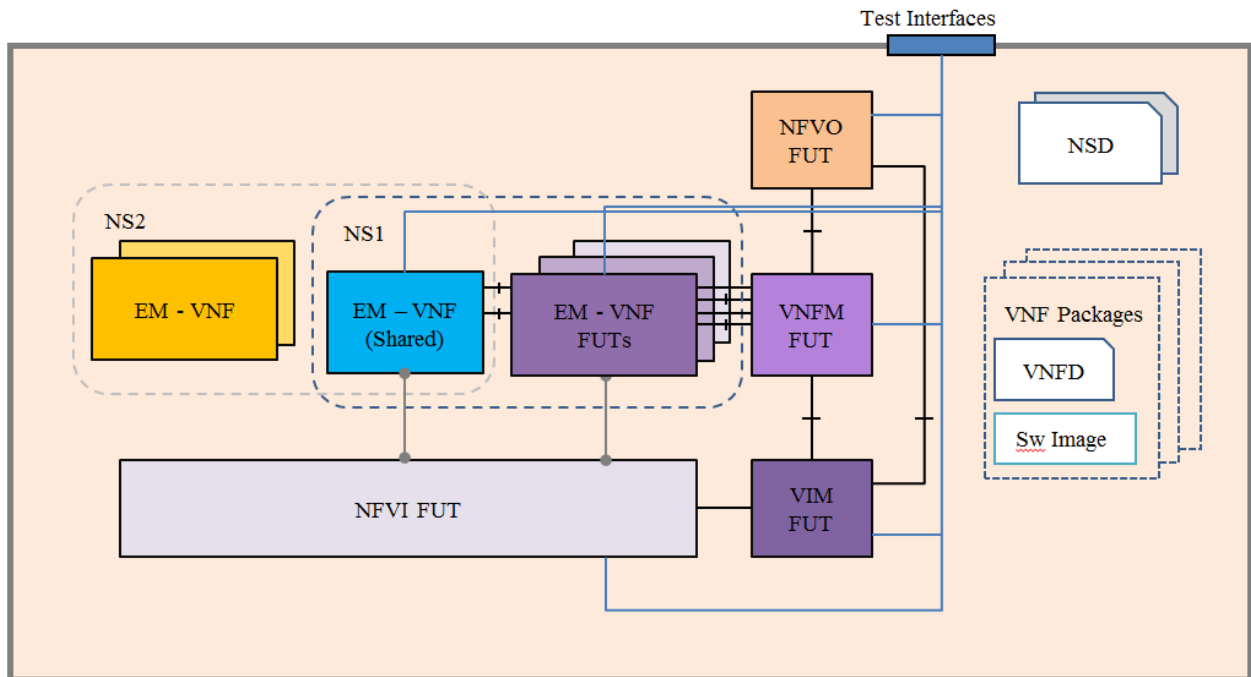


Figure 5.2-1: SUT Configuration 2

## 5.3 SUT Configuration 3

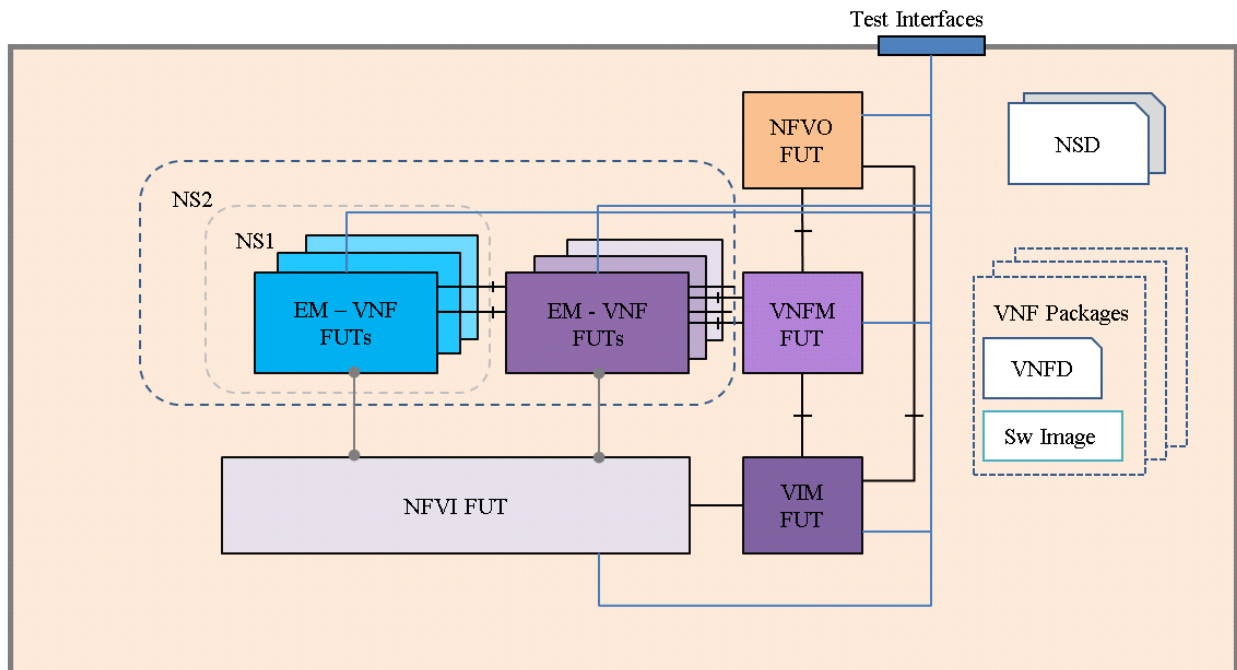


Figure 5.3-1: SUT Configuration 3

## 6 Test Suite Structure

### 6.1 Introduction

This clause describes the test suite structure and naming convention for the test descriptions according to ETSI GS NFV-TST 002 [i.1], clause 4.3.5.

The test cases in this test suite are grouped by the major functional areas they cover. The test identifiers use a common prefix of TD\_NFV\_ with an additional unique string for each area. Each test has a unique test identifier and a running number of the test case (in case of additional test case that cover similar test purpose).

The test case groups and their corresponding identifier naming prefix are described in table 6.1-1.

**Table 6.1-1**

Test Case Group	Identifier Name Prefix
Software Image Management	TD_NFV_SWIM_
VNF Package Management	TD_NFV_VNFPM_
VNF Lifecycle Management	TD_NFV_VNFLCM_
Fault Management	TD_NFV_FM_
Performance Management	TD_NFV_PM_
NS Lifecycle Management	TD_NFV_NSLCM_

### 6.2 Software Image Management Test Cases Overview

**Table 6.2-1**

Test description identifier	Test purpose
TD_NFV_SWIM_ADD_001	Verify that the NFVO can add a software image to the image repository managed by the VIM
TD_NFV_SWIM_QUERY_001	Verify that the NFVO can retrieve the information of a software image from the image repository managed by the VIM
TD_NFV_SWIM_QUERY_002	Verify that the VNFM can retrieve the information of a software image from the image repository managed by the VIM
TD_NFV_SWIM_UPDATE_001	Verify that the NFVO can update the metadata of a software image in the image repository managed by the VIM
TD_NFV_SWIM_DELETE_001	Verify that the NFVO can delete a software image from the image repository managed by the VIM

### 6.3 VNF Package Management Test Cases Overview

**Table 6.3-1**

Test case Identifier	Test case purpose
TD_NFV_VNFPM_ONBOARD_001	To verify that a VNF Package can be successfully on-boarded to the VNF catalogue managed by the NFVO
TD_NFV_VNFPM_DELETE_001	To verify that an on-boarded VNF Package can be successfully deleted from the VNF catalogue managed by the NFVO
TD_NFV_VNFPM_ABORTDEL_001	To verify that an on-boarded VNF Package that has been in deletion pending state can be successfully aborted the deletion by the NFVO

## 6.4 VNF Lifecycle Management Test Cases Overview

**Table 6.4-1**

Test case Identifier	Test case purpose
TD_NFV_VNFLCM_INSTANTIATE_VNF_001	Verify that a VNF be successfully instantiated when a "instantiate VNF" operation is triggered by the EM
TD_NFV_VNFLCM_QUERY_VNF_001	Verify that the VNF instance's information can be queried successfully by the EM
TD_NFV_VNFLCM_MODIFY_VNF_INFO_001	Verify that the VNF information can be updated successfully by the EM
TD_NFV_VNFLCM_START_VNF_001	Verify that a VNF/VNFC instance be successfully started when an "operate VNF" operation is triggered by the EM
TD_NFV_VNFLCM_STOP_VNF_001	Verify that a VNF/VNFC instance be successfully stopped when an "operate VNF" operation is triggered by the EM
TD_NFV_VNFLCM_HEAL_VNF_001	Verify that a VNF can be successfully healed when VNF healing request is sent out by the EM/VNF
TD_NFV_VNFLCM_SCALE_OUT_VNF_001	Verify that a VNF can be successfully scaled out by adding VNFC instances triggered by a EM/VNF request
TD_NFV_VNFLCM_SCALE_IN_VNF_001	Verify that a VNF can be successfully scaled in by removing VNFC instances triggered by a EM/VNF request
TD_NFV_VNFLCM_TERMINATE_VNF_001	Verify that a VNF be successfully terminated when a "terminate VNF" operation is triggered by the EM

## 6.5 Fault Management Test Cases Overview

**Table 6.5-1**

Test case Identifier	Test case purpose
TD_NFV_FM_VR_NOTIFY_001	Verify that a fault alarm notification propagates to the NFVO when a virtualised resource that is required for the NS connectivity fails
TD_NFV_FM_VR_CLEAR_001	Verify that a fault clearance notification propagates to the NFVO when a failed virtualised resource that is required for the NS connectivity is recovered
TD_NFV_FM_VNF_NOTIFY_001	Verify that a VNF fault alarm notification propagates via the VNFM to the NFVO when a VNF fault is triggered by a failed virtualised resource
TD_NFV_FM_VNF_CLEAR_001	Verify that a VNF fault alarm clearance notification propagates via the VNFM to the NFVO when a VNF fault is cleared by resolving a failed virtualised resource

## 6.6 Performance Management Test Cases Overview

**Table 6.6-1**

Test case Identifier	Test case purpose
TD_NFV_PM_VR_CREATE_NOTIFY_001	Verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and notifications
TD_NFV_PM_VR_CREATE_THRESHOLD_001	Verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and thresholds
TD_NFV_PM_VR_DELETE_MONITOR_001	Verify that the monitoring of performance metrics of a virtualised resource that is required for a NS instance connectivity can be stopped by deleting PM jobs
TD_NFV_PM_VR_DELETE_THRESHOLD_001	Verify that a threshold created for a virtualised resource that is required for a NS instance connectivity can be deleted
TD_NFV_PM_VNF_CREATE_NOTIFY_001	Verify that the performance metrics of a virtualised resource that is allocated to a VNF instance inside a NS instance can be monitored using PM jobs and notifications
TD_NFV_PM_VNF_CREATE_THRESHOLD_001	Verify that the performance metrics of a virtualised resource that is allocated to a VNF instance inside a NS instance can be monitored using PM jobs and thresholds
TD_NFV_PM_VNF_DELETE_MONITOR_001	Verify that the monitoring of performance metrics of a virtualised resource that is allocated to a VNF instance inside a NS instance can be stopped by deleting PM jobs
TD_NFV_PM_VNF_DELETE_THRESHOLD_001	Verify that a threshold created for a virtualised resource that is allocated to a VNF instance inside a NS instance can be deleted

## 6.7 NS Lifecycle Management Test Cases Overview

**Table 6.7-1**

Test case Identifier	Test case purpose
TD_NFV_NSLCM_INSTANTIATE_001	To verify that a standalone NS can be successfully instantiated
TD_NFV_NSLCM_INSTANTIATE_NEST_NS_001	To verify that a NS referencing an existing nested NS can be successfully instantiated
TD_NFV_NSLCM_SCALE_OUT_001	Verify that the NS can be successfully scaled out by adding VNF instances triggered by an operator action
TD_NFV_NSLCM_SCALE_OUT_002	Verify that the NS can be successfully scaled out by adding VNF instances triggered automatically by a VNF indicator
TD_NFV_NSLCM_SCALE_OUT_003	Verify that the NS can be successfully scaled out by adding VNF instances triggered automatically by a VIM KPI
TD_NFV_NSLCM_SCALE_IN_001	Verify that the NS can be successfully scaled in by removing VNF instances triggered by an operator action
TD_NFV_NSLCM_SCALE_IN_002	Verify that the NS can be successfully scaled in by removing VNF instances triggered automatically by a VNF indicator
TD_NFV_NSLCM_SCALE_IN_003	Verify that the NS can be successfully scaled in by removing VNF instances triggered automatically by a VIM KPI
TD_NFV_NSLCM_SCALE_OUT_VNF_001	To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered by a NFVO operator
TD_NFV_NSLCM_SCALE_OUT_VNF_002	To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered automatically by a VNF indicator
TD_NFV_NSLCM_SCALE_OUT_VNF_003	To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered automatically by a VIM KPI
TD_NFV_NSLCM_SCALE_IN_VNF_001	Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances from an existing VNF triggered by an operator action
TD_NFV_NSLCM_SCALE_IN_VNF_002	Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances triggered automatically by a VNF indicator
TD_NFV_NSLCM_SCALE_IN_VNF_003	Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances triggered automatically by a VIM KPI
TD_NFV_NSLCM_UPDATE_START_001	Verify the capability to start a VNF instance inside a NS instance
TD_NFV_NSLCM_UPDATE_STOP_001	Verify the capability to stop a VNF instance inside a NS instance
TD_NFV_NSLCM_UPDATE_INST_ADD_VNF_001	To verify that one or more VNFs can be instantiated and the instances added to a running NS instance
TD_NFV_NSLCM_UPDATE_REM_VNF_001	To verify that one or more VNF instances can be removed from a running NS instance
TD_NFV_NSLCM_UPDATE_ADD_SHVNF_001	To verify that one or more shared VNF instances can be added to a running NS instance
TD_NFV_NSLCM_UPDATE_REM_SHVNF_001	To verify that one or more shared VNF instances can be removed from a running NS instance
TD_NFV_NSLCM_UPDATE_ADD_VL_001	To verify that one or more virtual links (VL) can be added to a running NS instance
TD_NFV_NSLCM_UPDATE_REM_VL_001	To verify that one or more virtual links (VL) can be removed from a running NS instance
TD_NFV_NSLCM_UPDATE_VNF_DF_001	To verify that the deployment flavour of one or more VNF instances in a NS instance can be changed
TD_NFV_NSLCM_HEAL_001	Verify that VNF instances inside the NS can be successfully healed when partial NS healing (VNF healing) is triggered by an operator action
TD_NFV_NSLCM_HEAL_002	Verify that a NS can be successfully healed when complete NS healing is triggered by an operator action
TD_NFV_NSLCM_TERMINATE_001	To verify that a standalone NS instance can be successfully terminated
TD_NFV_NSLCM_TERMINATE_NESTED_NS_001	To verify that a NS instance referencing an existing nested NS can be successfully terminated

## 7 Test Descriptions

### 7.1 Introduction

This clause contains the test descriptions for the NFV functional areas. The results of IOP Checks and IOP Verdict inside each test description form should be indicated by OK/NOK according to clause 4.3.7 of ETSI GS NFV-TST 002 [i.1]. Non-applicable test descriptions can be marked as N/A. Collective results for functional areas can be reported according to the sample report card in annex A of the present document.

### 7.2 Software Image Management

#### 7.2.1 Add Software Image

Test Description: add software image				
<b>Identifier</b>	TD_NFV_SWIM_ADD_001			
<b>Test Purpose</b>	Verify that the NFVO can add a software image to the image repository managed by the VIM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can add software images on the VIM (NFVO_SWIM_ADD_IM)</li> <li>VIM supports "add image" operations by the NFVO (VIM_SWIM_ADD_IM)</li> <li>VIM supports software image information queries by the NFVO (VIM_SWIM_QUERY_IM_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>Software image information (e.g. name, metadata, location) is defined</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger an "add image" operation on the NFVO	
	2	IOP Check	Verify the image information by querying the list of images on the image repository managed by the VIM	
	3	IOP Check	Verify that the NFVO shows no "add image" operation errors	
<b>IOP Verdict</b>				

#### 7.2.2 Query Software Image

##### 7.2.2.1 Query Software Image by NFVO

Test Description: query software image by NFVO				
<b>Identifier</b>	TD_NFV_SWIM_QUERY_001			
<b>Test Purpose</b>	Verify that the NFVO can retrieve the information of a software image from the image repository managed by the VIM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can query software image information on the VIM (NFVO_SWIM_QUERY_IM)</li> <li>VIM supports software image information queries by the NFVO (VIM_SWIM_QUERY_IM_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The software image to be queried is added to the image repository managed by the VIM (TD_NFV_SWIM_ADD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the query of the relevant image information on the NFVO using the unique image identifier	
	2	IOP Check	Verify the image information by querying the list of images on the image repository managed by the VIM	
	3	IOP Check	Verify that the image information obtained by the NFVO matches the image information in Step2	
<b>IOP Verdict</b>				

## 7.2.2.2 Query Software Image by VNFM

Test Description: query software image by VNFM				
<b>Identifier</b>	TD_NFV_SWIM_QUERY_002			
<b>Test Purpose</b>	Verify that the VNFM can retrieve the information of a software image from the image repository managed by the VIM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• VNFM can query software image information on the VIM (VNFM_SWIM_QUERY_IM)</li> <li>• VIM supports software image information queries by the VNFM (VIM_SWIM_QUERY_IM_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• The software image to be queried is added to the image repository managed by the VIM (TD_NFV_SWIM_ADD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the query of the relevant image information on the VNFM using the unique image identifier	
	2	IOP Check	Verify the image information by querying the list of images on the image repository managed by the VIM	
	3	IOP Check	Verify that the image information obtained by the VNFM matches the image information in Step2	
<b>IOP Verdict</b>				

## 7.2.3 Update Software Image

Test Description: update software image				
<b>Identifier</b>	TD_NFV_SWIM_UPDATE_001			
<b>Test Purpose</b>	Verify that the NFVO can update the metadata of a software image in the image repository managed by the VIM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO can update software image information on the VIM (NFVO_SWIM_UPDATE_IM)</li> <li>• VIM supports "update image" operations by the NFVO (VIM_SWIM_UPDATE_IM)</li> <li>• VIM supports software image information queries by the NFVO (VIM_SWIM_QUERY_IM_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• The software image to be updated is added to the image repository managed by the VIM (TD_NFV_SWIM_ADD_001)</li> <li>• The updated software image metadata is defined</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger an "update image" operation on the NFVO using the unique image identifier and including the updated metadata	
	2	IOP Check	Verify the updated image information by querying the list of images on the image repository managed by the VIM	
	3	IOP Check	Verify that the NFVO shows no "update image" operation errors	
<b>IOP Verdict</b>				

## 7.2.4 Delete Software Image

Test Description: delete software image				
<b>Identifier</b>	TD_NFV_SWIM_DELETE_001			
<b>Test Purpose</b>	Verify that the NFVO can delete a software image from the image repository managed by the VIM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can delete software images from the VIM (NFVO_SWIM_DELETE_IM)</li> <li>VIM supports "delete image" operations by the NFVO (VIM_SWIM_DELETE_IM)</li> <li>VIM supports software image information queries by the NFVO (VIM_SWIM_QUERY_IM_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The software image to be deleted has been added to the image repository managed by the VIM (TD_NFV_SWIM_ADD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a "delete image" operation on the NFVO using the unique image identifier	
	2	IOP Check	Verify that the affected image has been deleted from the image repository managed by the VIM by querying the list of images	
	3	IOP Check	Verify that the NFVO shows no "delete image" operation errors	
<b>IOP Verdict</b>				

## 7.3 VNF Package Management

### 7.3.1 On-board VNF Package

Test Description: on-board VNF package				
<b>Identifier</b>	TD_NFV_VNFPM_ONBOARD_001			
<b>Test Purpose</b>	To verify that a VNF Package can be successfully on-boarded to the VNF catalogue managed by the NFVO			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 013 [i.9] and ETSI GS NFV-IFA 011 [i.8]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can check the integrity and validity of VNF Packages (NFVO_VNFPM_VALIDATE)</li> <li>NFVO can access the VNF catalogue where the VNF Packages information is stored (NFVO_VNFPM_ACCESS)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF Package contains all the necessary information including VNFD, software image and artefacts</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the on-boarding of a VNF Package on the NFVO	
	2	IOP Check	Verify that the VNF Package is stored in the VNF catalogue managed by the NFVO	
	3	IOP Check	Verify that a unique identifier for the VNF Package is generated by querying the VNF Package information on the NFVO	
<b>IOP Verdict</b>				



### 7.3.2 Delete VNF Package

Test Description: delete VNF package				
<b>Identifier</b>	TD_NFV_VNFPM_DELETE_001			
<b>Test Purpose</b>	To verify that an on-boarded VNF Package can be successfully deleted from the VNF catalogue managed by the NFVO			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can access the VNF catalogue where the VNF Packages information is stored (NFVO_VNFPM_ACCESS)</li> <li>NFVO supports on-boarded VNF Packages queries (NFVO_VNFPM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF Package is on-boarded, not in a "used" state, and not in a "deletion pending" state</li> <li>All the VNF instances associated to the VNF Package are terminated</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the deletion of a VNF Package on the NFVO	
	2	IOP Check	Verify that VNF Package has been deleted by querying the NFVO	
<b>IOP Verdict</b>				

### 7.3.3 Abort VNF Package Delete Operation

Test Description: abort VNF package delete operation				
<b>Identifier</b>	TD_NFV_VNFPM_ABORTDEL_001			
<b>Test Purpose</b>	To verify that an on-boarded VNF Package that is in deletion pending state can be successfully aborted from deletion by the NFVO			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO supports the delete VNF Package operation abortion (NFVO_VNFPM_ABORT_DELETE)</li> <li>NFVO supports on-boarded VNF Packages queries (NFVO_VNFPM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF Package is on-boarded and in "deletion pending" state</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the abortion of the deletion operation of an on-boarded VNF Package on NFVO	
	2	IOP Check	Verify that the VNF Package is not deleted by querying the NFVO	
	3	IOP Check	Verify the VNF Package is in disabled state by querying the NFVO	
<b>IOP Verdict</b>				

## 7.4 VNF Lifecycle Management

### 7.4.1 Instantiate VNF with an EM Request

Test Description: instantiate VNF with an EM request				
<b>Identifier</b>	TD_NFV_VNFLCM_INSTANTIATE_VNF_001			
<b>Test Purpose</b>	Verify that a VNF be successfully instantiated when an "instantiate VNF" operation is triggered by the EM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 008 [i.6]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can query software image information from the VIM (NFVO_SWIM_QUERY_IM or VNFM_SWIM_QUERY_IM)</li> <li>VIM supports software image information queries by the NFVO/VNFM (VIM_SWIM_QUERY_IM_BY_NFVO or VIM_SWIM_QUERY_IM_BY_VNFM)</li> <li>NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>NFVO/VNFM can query network resource information from the VIM (NFVO_NRM_QUERY or VNFM_NRM_QUERY)</li> <li>NFVO/VNFM can query storage resource information from the VIM (NFVO_SRM_QUERY or VNFM_SRM_QUERY)</li> <li>VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>EM can generate "Instantiate VNF" requests to the VNFM (EM_VNFLCM_VNF_INSTANTIATE)</li> <li>VNFM supports "instantiate VNF" requests from the EM (VNFM_VNFLCM_VNF_INSTANTIATE)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>Any constraints required for the virtualised resource such as affinity/anti-affinity rules and resource locations are defined in the VNFD</li> <li>The software image repository is reachable by the VIM</li> <li>The required resources are available on the NFVI</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger an "instantiate VNF" operation on the EM	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "instantiate VNF" operation has been approved by the NFVO	
	3	IOP Check	Verify that the software images have been successfully added to the image repository managed by the VIM	
	4	IOP Check	Verify that the requested virtualised resources have been allocated by the VIM according to the VNFD	
	5	IOP Check	Verify that virtualised resource allocation constraints have been met by querying the VIM	

<b>Test Sequence</b>	6	IOP Check	Verify that any existing virtualised resources have not been affected by the allocation of the new virtualised resources by querying the VIM	
	7	IOP Check	Verify that the VNF instance resources are visible on the VNFM	
	8	IOP Check	Verify that the VNF instance resources are visible on the NFVO	
	9	IOP Check	Verify that the VNF instance is reachable via the management network	
	10	IOP Check	Verify that the VNF instance has been configured according to the VNFD by querying the VNFM	
	11	IOP Check	Verify that the EM shows no "instantiate VNF" operation errors	
<b>IOP Verdict</b>				

## 7.4.2 Query VNF with an EM Request

<b>Test Description: query VNF information with an EM request</b>				
<b>Identifier</b>	TD_NFV_VNFLCM_QUERY_VNF_001			
<b>Test Purpose</b>	Verify that the VNF instance's information can be queried successfully by the EM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 008 [i.6]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>EM can generate "query VNF" requests to the VNFM (EM_VNFLCM_VNF_QUERY)</li> <li>VNFM supports "query VNF" requests from the EM (VNFM_VNFLCM_VNF_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF is instantiated (TD_NFV_VNFLCM_INSTANTIATE_VNF_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM to query the VNF information using the unique VNF instance identifier from the VNFM	
	2	IOP Check	Verify that the EM has received the VNF information from the VNFM	
	3	IOP Check	Verify that VNF information in Step 2 matches the VNF information on the VNFM	
	4	IOP Check	Verify that the EM indicates the "query VNF" operation result as successful	
<b>IOP Verdict</b>				

### 7.4.3 Modify VNF Configuration Information with an EM Request

Test Description: update VNF with an EM request				
<b>Identifier</b>	TD_NFV_VNFLCM_MODIFY_VNF_INFO_001			
<b>Test Purpose</b>	Verify that the VNF information can be updated successfully by the EM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 008 [i.6]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>EM can generate "modify VNF information" requests to the VNFM (EM_VNFLCM_VNF_MODIFY_INFO)</li> <li>VNFM supports "modify VNF information" requests from the EM (VNFM_VNFLCM_VNF_MODIFY_INFO)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF is instantiated (TD_NFV_VNFLCM_INSTANTIATE_VNF_001)</li> <li>The original VNF instance information is available for comparison (TD_NFV_VNFLCM_QUERY_VNF_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM to modify the VNF instance information on the VNFM	
	2	IOP Check	Verify that the VNF instance information has been updated on the VNFM	
	3	IOP Check	Verify that the VNF instance information has been updated by querying the NFVO	
	4	IOP Check	Verify that the EM indicates the "modify VNF information" operation result as successful	
<b>IOP Verdict</b>				

### 7.4.4 Start VNF/VNFC with an EM Request

Test Description: start VNF/VNFC with an EM request				
<b>Identifier</b>	TD_NFV_VNFLCM_START_VNF_001			
<b>Test Purpose</b>	Verify that a VNF/VNFC instance be successfully started when an "operate VNF" operation is triggered by the EM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 008 [i.6]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>EM can generate "operate VNF" requests to the VNFM (EM_VNFLCM_VNF_OPERATE)</li> <li>VNFM supports "operate VNF" requests from the EM (VNFM_VNFLCM_VNF_OPERATE)</li> <li>NFVO/VNFM can generate "operate compute resource" operation requests to the VIM (NFVO_CRM_OPERATE or VNFM_CRM_OPERATE)</li> <li>VIM supports "operate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_OPERATE_BY_NFVO or VIM_CRM_OPERATE_BY_VNFM)</li> <li>NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>VIM can send compute resource information to the NFVO/VNFM (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO/VNFM (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO/VNFM (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF is instantiated (TD_NFV_VNFLCM_INSTANTIATE_VNF_001)</li> <li>Target VNF/VNFC instance is in a "stopped" operational state. See note.</li> </ul>			

Test Description: start VNF/VNFC with an EM request				
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Trigger the EM to start the target VNF/VNFC instance	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "operate VNF" operation has been approved by the NFVO	
	3	IOP Check	Verify that the compute resources allocated to the target VNF/VNFC instance have been started by querying the VIM	
	4	IOP Check	Verify that other existing compute resources have not been affected by the performed operation by querying the VIM	
	5	IOP Check	Verify that the VNF/VNFC instance operational state on the VNFM is indicated as "started"	
	6	IOP Check	Verify that the EM indicates the "operate VNF" operation result as successful	
	7	IOP Check	Verify that the VNF/VNFC instance operates successfully by running the end-to-end functional test	
<b>IOP Verdict</b>				
NOTE: In the state STOPPED, the virtualised container(s), where the VNFC instance(s) of the VNF run, are shut down but not terminated.				

### 7.4.5 Stop VNF/VNFC with an EM Request

Test Description: stop VNF/VNFC with an EM request				
<b>Identifier</b>	TD_NFV_VNFLCM_STOP_VNF_001			
<b>Test Purpose</b>	Verify that a VNF/VNFC instance be successfully stopped when an "operate VNF" operation is triggered by the EM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 008 [i.6]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>EM can generate "operate VNF" requests to the VNFM (EM_VNFLCM_VNF_OPERATE)</li> <li>VNFM supports "operate VNF" requests from the EM (VNFM_VNFLCM_VNF_OPERATE)</li> <li>NFVO/VNFM can generate "operate compute resource" operation requests to the VIM (NFVO_CRM_OPERATE or VNFM_CRM_OPERATE)</li> <li>VIM supports "operate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_OPERATE_BY_NFVO or VIM_CRM_OPERATE_BY_VNFM)</li> <li>NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>VIM can send compute resource information to the NFVO/VNFM (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO/VNFM (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO/VNFM (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The VNF is instantiated (TD_NFV_VNFLCM_INSTANTIATE_VNF_001)</li> <li>Target VNF/VNFC instance is in a "started" operational state</li> </ul>			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Trigger the EM to stop the target VNF/VNFC instance	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "operate VNF" operation has been approved by the NFVO	
	3	IOP Check	Verify that the compute resources allocated to the target VNF/VNFC instance have been stopped by querying the VIM. See note.	
	4	IOP Check	Verify that other existing compute resources have not been affected by the performed operation by querying the VIM	

Test Description: stop VNF/VNFC with an EM request			
	5	IOP Check	Verify that the VNF/VNFC instance operational state on the VNFM is indicated as "stopped"
	6	IOP Check	Verify that the EM indicates the "operate VNF" operation result as successful
<b>IOP Verdict</b>			
NOTE: In the state STOPPED, the virtualised container(s), where the VNFC instance(s) of the VNF run, are shut down but not terminated.			

## 7.4.6 VNF Healing with an EM/VNF Request

Test Description: VNF healing with an EM/VNF Request				
<b>Identifier</b>	TD_NFV_VNFLCM_HEAL_VNF_001			
<b>Test Purpose</b>	Verify that a VNF can be successfully healed when a VNF healing request is sent out by the EM/VNF			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>VNFM supports "heal VNF" requests from EM/VNF (VNFM_VNFLCM_VNF_HEAL)</li> <li>EM/VNF can generate "heal VNF" requests to the VNFM (EM_VNFLCM_VNF_HEAL)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>VNF is in a failed state (for example a virtualised resource needed by the impacted VNF instance has been terminated directly on the VIM)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM/VNF to send a VNF healing request to the VNFM	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "heal VNF" operation has been approved by the NFVO	
	3	IOP Check	Verify that any additional resources required for the healing process have been allocated by the VIM according to the descriptors	
	4	IOP Check	Verify that healed VNF instance is running and reachable via the management network	
	5	IOP Check	Verify that the healed VNF instance has been configured	

Test Description: VNF healing with an EM/VNF Request				
			according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that any failed resources have been terminated and released by the VIM	
	7	IOP Check	Verify that the EM/VNF indicates the VNF healing operation result as successful	
	8	IOP Check	Verify that the NFVO indicates the VNF healing operation result as successful	
	9	IOP Check	Verify that the VNF has been healed by running an end-to-end functional test factoring in the functionality of the healed VNF instance	
<b>IOP Verdict</b>				

#### 7.4.7 VNF Scale Out with an EM/VNF Request

Test Description: scale out VNF with an EM/VNF request				
<b>Identifier</b>	TD_NFV_VNFLCM_SCALE_OUT_VNF_001			
<b>Test Purpose</b>	Verify that a VNF can be successfully scaled out by adding VNFC instances triggered by the EM/VNF			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>EM/VNF can generate "scale out by adding VNFC instances" requests to the VNFM (EM_VNFLCM_VNF_SCALE_OUT)</li> <li>VNFM supports "scale out by adding VNFC instances" requests from the EM/VNF (VNFM_VNFLCM_VNF_SCALE_OUT)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>VNFM is configured to trigger scale out when scale out request is received from EM/VNF</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out VNF</li> <li>The current scale level of the VNF instance is not at the maximum scale level</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM/VNF to send a "scale out by adding VNFC instance(s)" request to the VNFM	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "VNF scale out" operation has been approved by the NFVO	
	3	IOP Check	Verify that the "scale out by adding VNFC instance(s)" procedure has been started on the NFVO	
	4	IOP Check	Verify that the additional VNFC instance(s) have been deployed by querying the VNFM	
	5	IOP Check	Verify that the additional resources have been allocated by the VIM according to the VNFD	
	6	IOP Check	Verify that the additional VNFC instance(s) are running and are reachable through their management network	
	7	IOP Check	Verify that the VNF configuration has been updated to include the additional VNFC instances according to the VNFD by querying the VNFM	
	8	IOP Check	Verify that the additional VNFC instances(s) are connected to	

Test Description: scale out VNF with an EM/VNF request			
			the VL(s) according to the VNFD
	9	IOP Check	Verify that the EM/VNF indicates the scaling operation result as successful
	10	IOP Check	Verify that the NFVO indicates the scaling operation result as successful
	11	IOP Check	Verify that VNF has been scaled out by running the end-to-end functional test in relevance to the VNF scale and capacity
<b>IOP Verdict</b>			

## 7.4.8 VNF Scale In with an EM/VNF Request

Test Description: VNF scale in with an EM/VNF request				
<b>Identifier</b>	TD_NFV_VNFLCM_SCALE_IN_VNF_001			
<b>Test Purpose</b>	Verify that a VNF can be successfully scaled in by removing VNFC instances triggered by the EM/VNF			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>EM/VNF can generate "scale in by removing VNFC instances" to the VNFM (EM_VNFLCM_VNF_SCALE_IN)</li> <li>VNFM supports "scale in by removing VNFC instances" requests from the EM/VNF (VNFM_VNFLCM_VNF_SCALE_IN)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The current NS deployment size allows scaling in</li> <li>The current scale level of the VNF instance is not at the minimum scale level</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM/VNF to send a "scale in by removing VNFC instance(s)" request to the VNFM	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "VNF scale in" operation has been approved by the NFVO	
	3	IOP Check	Verify that the "scale in by removing VNFC instance(s)" procedure has been started in NFVO	
	4	IOP Check	Verify that the impacted VNFC instance(s) inside the VNF have been terminated by querying the VNFM	
	5	IOP Check	Verify that the impacted VNFC instance(s) resources have been released by the VIM	
	6	IOP Check	Verify that the remaining VNFC instance(s) are still running and reachable via their management network	
	7	IOP Check	Verify that the VNF configuration has been updated to exclude the removed VNFC instances according to the descriptors by querying the VNFM	
	8	IOP Check	Verify that the remaining VNFC instances(s) and VL(s) are still connected according to the descriptors	
	9	IOP Check	Verify that the EM/VNF indicates the scaling operation result as successful	
	10	IOP Check	Verify that the NFVO indicates the scaling operation result as	



Test Description: VNF scale in with an EM/VNF request			
			successful
	11	IOP Check	Verify that the VNF has been scaled in by running the end-to-end functional test in relevance to the VNF scale and capacity
<b>IOP Verdict</b>			

## 7.4.9 Terminate VNF with an EM Request

Test Description: terminate VNF with an EM request				
<b>Identifier</b>	TD_NFV_VNFLCM_TERMINATE_VNF_001			
<b>Test Purpose</b>	Verify that a VNF is successfully terminated when a "terminate VNF" operation is triggered by the EM			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 008 [i.6]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>NFVO/VNFM can query network resource information from the VIM (NFVO_NRM_QUERY or VNFM_NRM_QUERY)</li> <li>NFVO/VNFM can query storage resource information from the VIM (NFVO_SRM_QUERY or VNFM_SRM_QUERY)</li> <li>VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>EM can generate "terminate VNF" requests to the VNFM (EM_VNFLCM_VNF_TERMINATE)</li> <li>VNFM supports "terminate VNF" requests from the EM (VNFM_VNFLCM_VNF_TERMINATE)</li> <li>Optionally, VNFM can generate "Grant VNF Lifecycle Operation" requests to the NFVO (VNFM_VNFLCM_GRANTING) and NFVO supports "Grant VNF Lifecycle Operation" requests from the VNFM (NFVO_VNFLCM_GRANTING)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>The target VNF is instantiated</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger an "terminate VNF" operation on the EM	
	2	IOP Check	If VNFM_VNFLCM_GRANTING and NFVO_VNFLCM_GRANTING are supported, verify that the requested grant for the "VNF terminate" operation has been approved by the NFVO	
	3	IOP Check	Verify that the impacted virtualised resources have been terminated by the VIM	
	4	IOP Check	Verify that other allocated virtualised resources have not been affected by the termination of the virtualised resources of the VNF instance by querying the VIM	
<b>IOP Verdict</b>				

## 7.5 Fault Management

### 7.5.1 Virtualised Resource Fault Management

#### 7.5.1.1 Virtualised Resource Fault Alarm Notification

Test Description: virtualised resource fault alarm notification				
<b>Identifier</b>	TD_NFV_FM_VR_NOTIFY_001			
<b>Test Purpose</b>	Verify that a fault alarm notification propagates to the NFVO when a virtualised resource that is required for the NS connectivity fails			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can subscribe to virtualised resource fault alarms on the VIM (NFVO_FM_VR_SUBSCRIBE)</li> <li>VIM supports alarm notifications subscriptions from the NFVO (VIM_FM_SUBSCRIBE_BY_NFVO)</li> <li>VIM can generate virtualised resources fault alarm notifications to the NFVO (VIM_FM_NOTIFY_BY_NFVO)</li> <li>NFVO can process virtualised resource fault alarm notifications from the VIM (NFVO_FM_VR_NOTIFY)</li> <li>NFVO can get the list of virtualised resource fault alarms from the VIM (NFVO_FM_VR_GET_ALARMS)</li> <li>VIM supports virtualised resource fault alarm list queries by the NFVO (VIM_FM_QUERY_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVO is subscribed to virtualised resources fault alarms on the VIM</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a fault on a virtualised resource that is required for the NS instance connectivity on the NFVI (e.g. disable the NIC allocated to a network resource)	
	2	IOP Check	Verify that a virtualised resource fault alarm has been created on the VIM by querying the list of virtualised resource fault alarms	
	3	IOP Check	Verify that a NS fault alarm has been created on the NFVO by querying the list of NS fault alarms	
<b>IOP Verdict</b>				

## 7.5.1.2 Virtualised Resource Fault Alarm Clearance Notification

<b>Test Description: virtualised resource fault alarm clearance notification</b>				
<b>Identifier</b>	TD_NFV_FM_VR_CLEAR_001			
<b>Test Purpose</b>	Verify that a fault clearance notification propagates to the NFVO when a failed virtualised resource that is required for the NS connectivity is recovered			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO can subscribe to virtualised resource fault alarms on the VIM (NFVO_FM_VR_SUBSCRIBE)</li> <li>• VIM supports virtualised resource fault alarms subscriptions from the NFVO (VIM_FM_SUBSCRIBE_BY_NFVO)</li> <li>• VIM can generate virtualised resource fault alarm notifications to the NFVO (VIM_FM_NOTIFY_BY_NFVO)</li> <li>• NFVO can process virtualised resource fault alarm notifications from the VIM (NFVO_FM_VR_NOTIFY)</li> <li>• NFVO can get the list of virtualised resource fault alarms from the VIM (NFVO_FM_VR_GET_ALARMS)</li> <li>• VIM supports virtualised resource fault alarm list queries by the NFVO (VIM_FM_QUERY_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>• NFVO is subscribed to virtualised resources fault alarms on the VIM</li> <li>• NS fault alarm is created on the NFVO by failing a virtualised resource that is required for the NS connectivity (TD_NFV_FM_VR_NOTIFY_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Resolve the failure of the virtualised resource that is required for the NS instance connectivity (e.g. reconnect the NIC allocated to a network resource)	
	2	IOP Check	Verify that the relevant virtualised resource fault alarm has been cleared on the VIM by querying the list of virtualised resource fault alarms	
	3	IOP Check	Verify that the relevant NS fault alarm has been cleared on the NFVO by querying the list of NS fault alarms	
<b>IOP Verdict</b>				

## 7.5.2 VNF Fault Management

### 7.5.2.1 VNF Fault Alarm Notifications

Test Description: VNF fault alarm notification				
<b>Identifier</b>	TD_NFV_FM_VNF_NOTIFY_001			
<b>Test Purpose</b>	Verify that a VNF fault alarm notification propagates via the VNFM to the NFVO when a VNF fault is triggered by a failed virtualised resource			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO can subscribe to alarm notifications on the VNFM (NFVO_FM_VNF_SUBSCRIBE)</li> <li>• VNFM supports alarm notifications subscriptions from the NFVO (VNFM_FM_VNF_SUBSCRIBE)</li> <li>• VNFM can subscribe to alarm notifications on the VIM (VNFM_FM_VR_SUBSCRIBE)</li> <li>• VIM supports alarm notifications subscriptions by the VNFM (VIM_FM_SUBSCRIBE_BY_VNFM)</li> <li>• VNFM can generate VNF fault alarm notifications to the NFVO (VNFM_FM_VNF_NOTIFY)</li> <li>• NFVO can process VNF fault alarm notifications from the VNFM (NFVO_FM_VNF_NOTIFY)</li> <li>• VIM can generate virtualised resources fault alarm notifications to the VNFM (VIM_FM_NOTIFY_BY_VNFM)</li> <li>• VNFM can process virtualised resource fault alarm notifications by the VIM (VNFM_FM_VR_NOTIFY)</li> <li>• NFVO can get the list of VNF fault alarms from the VNFM (NFVO_FM_VNF_GET_ALARMS)</li> <li>• VNFM supports VNF fault alarm list queries by the NFVO (VNFM_FM_VNF_QUERY)</li> <li>• VNFM can get the list of virtualised resource fault alarms from the VIM (VNFM_FM_VR_GET_ALARM)</li> <li>• VIM supports virtualised resources fault alarm list queries by the VNFM (VIM_FM_QUERY_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>• NFVO is subscribed to VNF fault alarms on the VNFM</li> <li>• VNFM is subscribed to virtualised resources fault alarms on the VIM</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a failure on a virtualised resource allocated to the relevant VNF instance (e.g. terminate the virtualised resource directly on the VIM)	
	2	IOP Check	Verify that a virtualised resource fault alarm has been created on the VIM by querying the list of virtualised resource fault alarms	
	3	IOP Check	Verify that a VNF fault alarm has been created for the affected VNF instance on the VNFM by querying the list of VNF fault alarms	
	4	IOP Check	Verify that a NS fault alarm has been created on the NFVO by querying the list of NS fault alarms	
<b>IOP Verdict</b>				

## 7.5.2.2 VNF Fault Alarm Clearance Notifications

<b>Test Description: VNF fault alarm clearance notification</b>				
<b>Identifier</b>	TD_NFV_FM_VNF_CLEAR_001			
<b>Test Purpose</b>	Verify that a VNF fault alarm clearance notification propagates via the VNFM to the NFVO when a VNF fault is cleared by resolving a failed virtualised resource			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO can subscribe to alarm notifications on the VNFM (NFVO_FM_VNF_SUBSCRIBE)</li> <li>• VNFM supports alarm notifications subscriptions from the NFVO (VNFM_FM_VNF_SUBSCRIBE)</li> <li>• VNFM can subscribe to alarm notifications on the VIM (VNFM_FM_VR_SUBSCRIBE)</li> <li>• VIM supports alarm notifications subscriptions by the VNFM (VIM_FM_SUBSCRIBE_BY_VNFM)</li> <li>• VNFM can generate VNF fault alarm clearance notifications to the NFVO (VNFM_FM_VNF_NOTIFY)</li> <li>• NFVO can process VNF fault alarm clearance notifications from the VNFM (NFVO_FM_VNF_NOTIFY)</li> <li>• VIM can generate virtualised resources fault alarm clearance notifications to the VNFM (VIM_FM_NOTIFY_BY_VNFM)</li> <li>• VNFM can process virtualised resource fault alarm clearance notifications from the VIM (VNFM_FM_VR_NOTIFY)</li> <li>• NFVO can get the list of VNF fault alarms from the VNFM (NFVO_FM_VNF_GET_ALARM)</li> <li>• VNFM supports VNF fault alarm list queries by the NFVO (VNFM_FM_VNF_QUERY)</li> <li>• VIM supports virtualised resources fault alarm list queries by the VNFM (VIM_FM_QUERY_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>• NFVO is subscribed to VNF fault alarms on the VNFM</li> <li>• VNFM is subscribed to virtualised resources fault alarms on the VIM</li> <li>• NS fault alarm is created on the NFVO by failing a virtualised resource that is allocated to the relevant VNF instance (TD_NFV_FM_VNF_NOTIFY_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Resolve the failure of the virtualised resource allocated to the relevant VNF (e.g. restart the virtualised resource directly on the VIM)	
	2	IOP Check	Verify that the relevant virtualised resource fault alarm has been cleared on the VIM by querying the list of virtualised resource fault alarms	
	3	IOP Check	Verify that the relevant VNF fault alarm has been cleared on the VNFM by querying the list of VNF fault alarms	
	4	IOP Check	Verify that the relevant NS fault alarm has been cleared on the NFVO by querying the list of NS fault alarms	
<b>IOP Verdict</b>				

## 7.6 Performance Management

### 7.6.1 Virtualised Resource Performance Management

#### 7.6.1.1 Virtualised Resource PM Job Creation and Notification Monitoring

Test Description: VR PM job creation and notification monitoring				
<b>Identifier</b>	TD_NFV_PM_VR_CREATE_NOTIFY_001			
<b>Test Purpose</b>	Verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and notifications			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO can create VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_CREATE)</li> <li>• VIM supports VR PM jobs creation by the NFVO (VIM_PM_PMJOB_CREATE_BY_NFVO)</li> <li>• NFVO can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE)</li> <li>• VIM supports VR PM job subscriptions from the NFVO (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO)</li> <li>• VIM can generate VR PM notifications to the NFVO (VIM_PM_PMJOB_NOTIFY_BY_NFVO)</li> <li>• NFVO supports VR PM job notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY)</li> <li>• NFVO can query VR PM jobs from the VIM (NFVO_PM_VR_PMJOB_QUERY)</li> <li>• VIM supports VR PM job queries by the NFVO (VIM_PM_PMJOB_QUERY_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>• Monitoring parameters (e.g. metrics, metric groups, collection and reporting periods) are defined</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to create a VR PM job for a virtualised resource that is allocated to the target NS instance	
	2	IOP Check	Verify that a VR PM job has been created on the VIM according to the monitoring parameters by querying the VR PM jobs	
	3	Stimulus	Trigger the NFVO to subscribe to the VR PM job created in step 1	
	4	IOP Check	Verify that a "performance information available" notification for the monitored virtualised resource was generated by the VIM to the NFVO by monitoring the VR PM notifications. See note.	
<b>IOP Verdict</b>				
NOTE: Metric collection and reporting periods should be taken into account.				

## 7.6.1.2 Virtualised Resource PM Job Creation and Threshold Monitoring

Test Description: VR PM job creation and threshold monitoring				
<b>Identifier</b>	TD_NFV_PM_VR_CREATE_THRESHOLD_001			
<b>Test Purpose</b>	Verify that the performance metrics of a virtualised resource that is required for a NS instance connectivity can be monitored using PM jobs and thresholds			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can create VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the NFVO (VIM_PM_PMJOB_CREATE_BY_NFVO)</li> <li>NFVO can create VR PM thresholds on the VIM (NFVO_PM_VR_PMTH_CREATE)</li> <li>VIM supports VR PM thresholds creation by the NFVO (VIM_PM_PMTH_CREATE_BY_NFVO)</li> <li>VIM can generate VR PM notifications to the NFVO (VIM_PM_PMJOB_NOTIFY_BY_NFVO)</li> <li>NFVO supports VR PM job notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can query VR PM jobs from the VIM (NFVO_PM_VR_PMJOB_QUERY)</li> <li>VIM supports VR PM job queries by the NFVO (VIM_PM_PMJOB_QUERY_BY_NFVO)</li> <li>NFVO can query VR PM thresholds from the VIM (NFVO_PM_VR_PMTH_QUERY)</li> <li>VIM supports VR PM threshold queries by the NFVO (VIM_PM_PMTH_QUERY_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>Monitoring parameters (e.g. metrics, metric groups, thresholds) are defined</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to create a VR PM job for a virtualised resource that is allocated to the target NS instance	
	2	IOP Check	Verify that a VR PM job has been created on the VIM according to the monitoring parameters by querying the VR PM jobs	
	3	Stimulus	Trigger the NFVO to create a VR PM threshold for the virtualised resource monitored in step 1	
	4	Stimulus	Trigger the NFVO to subscribe to the threshold crossing notification for the VR PM threshold created in step 3	
	5	Stimulus	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container)	
	6	IOP Check	Verify that the "threshold crossed" notification for the monitored virtualised resource was generated by the VIM to the NFVO by monitoring the VR PM notifications	
<b>IOP Verdict</b>				

## 7.6.1.3 Virtualised Resource PM Job Deletion

Test Description: VR PM job deletion				
<b>Identifier</b>	TD_NFV_PM_VR_DELETE_MONITOR_001			
<b>Test Purpose</b>	Verify that the monitoring of performance metrics of a virtualised resource that is required for a NS instance connectivity can be stopped by deleting PM jobs			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can delete VR PM jobs from the VIM (NFVO_PM_VR_PMJOB_DELETE)</li> <li>VIM supports VR PM jobs deletion by the NFVO (VIM_PM_PMJOB_DELETE_BY_NFVO)</li> <li>NFVO can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the NFVO (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO)</li> <li>VIM can generate VR PM notifications to the NFVO (VIM_PM_PMJOB_NOTIFY_BY_NFVO)</li> <li>NFVO supports VR PM job notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can query VR PM jobs from the VIM (NFVO_PM_VR_PMJOB_QUERY)</li> <li>VIM supports VR PM job queries by the NFVO (VIM_PM_PMJOB_QUERY_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>A virtualised resource that is required for the NS connectivity is monitored by the NFVO (TD_NFV_PM_VR_CREATE_NOTIFY_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to delete a VR PM job of a virtualised resource that is allocated to the target NS instance	
	2	IOP Check	Verify that the relevant VR PM job has been deleted on the VIM by querying the VR PM jobs	
	3	IOP Check	Verify that no "performance information available" notification for the monitored virtualised resource has been generated by the VIM to the NFVO by monitoring the VR PM notifications. See note.	
<b>IOP Verdict</b>				
NOTE: Virtualised resource metric collection and reporting periods should be taken into account.				

## 7.6.1.4 Virtualised Resource PM Threshold Deletion

Test Description: VR PM threshold deletion				
<b>Identifier</b>	TD_NFV_PM_VR_DELETE_THRESHOLD_001			
<b>Test Purpose</b>	Verify that a threshold created for a virtualised resource that is required for a NS instance connectivity can be deleted			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can delete VR PM thresholds from the VIM (NFVO_PM_VR_PMTN_DELETE)</li> <li>VIM supports VR PM thresholds deletion by the NFVO (VIM_PM_PMTN_DELETE_BY_NFVO)</li> <li>NFVO can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the NFVO (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO)</li> <li>VIM can generate VR PM notifications to the NFVO (VIM_PM_PMJOB_NOTIFY_BY_NFVO)</li> <li>NFVO supports VR PM job notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can query VR PM thresholds from the VIM (NFVO_PM_VR_PMTN_QUERY)</li> <li>VIM supports VR PM threshold queries by the NFVO (VIM_PM_PMTN_QUERY_BY_NFVO)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>A threshold for a virtualised resource that is required for the NS connectivity is created by the NFVO (TD_NFV_PM_VR_CREATE_THRESHOLD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to delete a VR PM threshold of a virtualised resource that is allocated to the target NS instance	
	2	IOP Check	Verify that the relevant VR PM threshold has been deleted on the VIM by querying the VR PM thresholds	



	3	Stimulus	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container)	
	4	IOP Check	Verify that no "threshold crossed" notification for the monitored virtualised resource has been generated by the VIM to the NFVO by monitoring the VR PM notifications. See note.	
<b>IOP Verdict</b>				
NOTE: Virtualised resource metric collection and reporting periods should be taken into account.				

## 7.6.2 VNF Performance Management

### 7.6.2.1 VNF PM Job Creation and Notification Monitoring

Test Description: VNF PM job creation and notification monitoring				
<b>Identifier</b>	TD_NFV_PM_VNF_CREATE_NOTIFY_001			
<b>Test Purpose</b>	Verify that the performance metrics of a virtualised resource that is allocated to a VNF instance inside a NS instance can be monitored using PM jobs and notifications			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can create VNF PM jobs on the VNFM (NFVO_PM_VNF_PMJOB_DELETE)</li> <li>VNFM supports VNF PM jobs creation by the NFVO (VNFM_PM_VNF_PMJOB_CREATE)</li> <li>VNFM can create VR PM jobs on the VIM (VNFM_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the VNFM (VIM_PM_PMJOB_CREATE_BY_VNFM)</li> <li>NFVO can subscribe to VNF PM jobs on the VNFM (NFVO_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM supports VNF PM job subscriptions from the NFVO (VNFM_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM can subscribe to VR PM jobs on the VIM (VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the VNFM (VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>VNFM supports VR PM job notifications from the VIM (VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>VNFM can generate VNF PM notifications to the NFVO (VNFM_PM_VNF_PMJOB_NOTIFY)</li> <li>NFVO supports VNF PM job notifications from the VNFM (NFVO_PM_VNF_PMJOB_NOTIFY)</li> <li>VNFM can query VR PM jobs from the VIM (VNFM_PM_VR_PMJOB_QUERY)</li> <li>VIM supports VR PM job queries by the VNFM (VIM_PM_PMJOB_QUERY_BY_VNFM)</li> <li>NFVO can query VNF PM jobs from the VNFM (NFVO_PM_VNF_PMJOB_QUERY)</li> <li>VNFM supports VNF PM job queries by the NFVO (VNFM_PM_VNF_PMJOB_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>Monitoring parameters (e.g. metrics, metric groups, collection and reporting periods) are defined on NFVO</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to create a VNF PM job for a virtualised resource allocated to the relevant VNF instance inside the target NS instance	
	2	IOP Check	Verify that a VNF PM job has been created on the VNFM according to the monitoring parameters by querying the VNF PM jobs	
	3	IOP Check	Verify that a virtualised resource PM job has been created on the VIM according to the monitoring parameters by querying the VR PM jobs	
	4	Stimulus	Trigger the NFVO to subscribe to the VNF PM job created in step 1	
	5	IOP Check	Verify that a "performance information available" notification for the monitored virtualised resource was generated by the VIM to the VNFM by monitoring the VR PM notifications. See note.	
	6	IOP Check	Verify that a "performance information available" notification for the monitored virtualised resource was generated by the VNFM to the NFVO by monitoring the VNF PM notifications. See note.	

Test Description: VNF PM job creation and notification monitoring	
IOP Verdict	
NOTE: Metric collection and reporting periods should be taken into account.	

### 7.6.2.2 VNF PM Job Creation and Threshold Monitoring

Test Description: VNF PM job creation and threshold monitoring																													
Identifier	TD_NFV_PM_VNF_CREATE_THRESHOLD_001																												
Test Purpose	Verify that the performance metrics of a virtualised resource that is allocated to a VNF instance inside a NS instance can be monitored using PM jobs and thresholds																												
Configuration	SUT Configuration 1																												
References	ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]																												
Applicability	<ul style="list-style-type: none"> <li>NFVO can create VNF PM jobs on the VNFM (NFVO_PM_VNF_PMJOB_CREATE)</li> <li>VNFM supports VNF PM jobs creation by the NFVO (VNFM_PM_VNF_PMJOB_CREATE)</li> <li>VNFM can create VR PM jobs on the VIM (VNFM_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the VNFM (VIM_PM_PMJOB_CREATE_BY_VNFM)</li> <li>NFVO can create VNF PM thresholds on the VNFM (NFVO_PM_VNF_PMTH_CREATE)</li> <li>VNFM supports VNF PM thresholds creation by the NFVO (VNFM_PM_VNF_PMTH_CREATE)</li> <li>VNFM can create VR PM thresholds on the VIM (VNFM_PM_VR_PMTH_CREATE)</li> <li>VIM supports VR PM thresholds creation by the VNFM (VIM_PM_PMTH_CREATE_BY_VNFM)</li> <li>NFVO can subscribe to VNF PM jobs on the VNFM (NFVO_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM supports VNF PM job subscriptions from the NFVO (VNFM_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM can subscribe to VR PM jobs on the VIM (VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the VNFM (VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>VNFM supports VR PM job notifications from the VIM (VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>VNFM can generate VNF PM notifications to the NFVO (VNFM_PM_VNF_PMJOB_NOTIFY)</li> <li>NFVO supports VNF PM job notifications from the VNFM (NFVO_PM_VNF_PMJOB_NOTIFY)</li> <li>VNFM can query VR PM jobs from the VIM (VNFM_PM_VR_PMJOB_QUERY)</li> <li>VIM supports VR PM job queries by the VNFM (VIM_PM_PMJOB_QUERY_BY_VNFM)</li> <li>NFVO can query VNF PM jobs from the VNFM (NFVO_PM_VNF_PMJOB_QUERY)</li> <li>VNFM supports VNF PM job queries by the NFVO (VNFM_PM_VNF_PMJOB_QUERY)</li> <li>VNFM can query VR PM thresholds from the VIM (VNFM_PM_VR_PMTH_QUERY)</li> <li>VIM supports VR PM threshold queries by the VNFM (VIM_PM_PMTH_QUERY_BY_VNFM)</li> <li>NFVO can query VNF PM thresholds from the VNFM (NFVO_PM_VNF_PMTH_QUERY)</li> <li>VNFM supports VNF PM threshold queries by the NFVO (VNFM_PM_VNF_PMTH_QUERY)</li> </ul>																												
Pre-test conditions	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>Monitoring parameters (e.g. VNF instance, metrics, metric groups, threshold) are defined on NFVO</li> </ul>																												
<b>Test Sequence</b>	<table border="1"> <thead> <tr> <th>Step</th> <th>Type</th> <th>Description</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Stimulus</td> <td>Trigger the NFVO to create a PM job for a virtualised resource allocated to the relevant VNF instance inside the target NS instance</td> <td></td> </tr> <tr> <td>2</td> <td>IOP Check</td> <td>Verify that a VNF PM job has been created on the VNFM according to the monitoring parameters by querying the VNF PM jobs</td> <td></td> </tr> <tr> <td>3</td> <td>IOP Check</td> <td>Verify that a VR PM job has been created on the VIM according to the monitoring parameters by querying the VR PM jobs</td> <td></td> </tr> <tr> <td>4</td> <td>Stimulus</td> <td>Trigger the NFVO to create a VNF PM threshold for the virtualised resource monitored in step 1</td> <td></td> </tr> <tr> <td>5</td> <td>Stimulus</td> <td>Trigger the NFVO to subscribe to the threshold crossing notification for the VNF PM threshold created in step 4</td> <td></td> </tr> <tr> <td>6</td> <td>Stimulus</td> <td>Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container)</td> <td></td> </tr> </tbody> </table>	Step	Type	Description	Result	1	Stimulus	Trigger the NFVO to create a PM job for a virtualised resource allocated to the relevant VNF instance inside the target NS instance		2	IOP Check	Verify that a VNF PM job has been created on the VNFM according to the monitoring parameters by querying the VNF PM jobs		3	IOP Check	Verify that a VR PM job has been created on the VIM according to the monitoring parameters by querying the VR PM jobs		4	Stimulus	Trigger the NFVO to create a VNF PM threshold for the virtualised resource monitored in step 1		5	Stimulus	Trigger the NFVO to subscribe to the threshold crossing notification for the VNF PM threshold created in step 4		6	Stimulus	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container)	
Step	Type	Description	Result																										
1	Stimulus	Trigger the NFVO to create a PM job for a virtualised resource allocated to the relevant VNF instance inside the target NS instance																											
2	IOP Check	Verify that a VNF PM job has been created on the VNFM according to the monitoring parameters by querying the VNF PM jobs																											
3	IOP Check	Verify that a VR PM job has been created on the VIM according to the monitoring parameters by querying the VR PM jobs																											
4	Stimulus	Trigger the NFVO to create a VNF PM threshold for the virtualised resource monitored in step 1																											
5	Stimulus	Trigger the NFVO to subscribe to the threshold crossing notification for the VNF PM threshold created in step 4																											
6	Stimulus	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container)																											

Test Description: VNF PM job creation and threshold monitoring			
	7	IOP Check	Verify that the "threshold crossed" notification for the monitored virtualised resource was generated by the VIM to the VNFM by monitoring the VR PM notifications
	8	IOP Check	Verify that the "threshold crossed" notification for the monitored virtualised resource was generated by the VNFM to the NFVO by monitoring the VNF PM notifications
IOP Verdict			

### 7.6.2.3 VNF PM Job Deletion

Test Description: VNF PM job deletion				
Identifier	TD_NFV_PM_VNF_DELETE_MONITOR_001			
Test Purpose	Verify that the monitoring of performance metrics of a virtualised resource that is allocated to a VNF instance inside a NS instance can be stopped by deleting PM jobs			
Configuration	SUT Configuration 1			
References	ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
Applicability	<ul style="list-style-type: none"> <li>NFVO can delete VNF PM jobs from the VNFM (NFVO_PM_VNF_PMJOB_DELETE)</li> <li>VNFM supports VNF PM jobs deletion by the NFVO (VNFM_PM_VNF_PMJOB_DELETE)</li> <li>VNFM can delete VR PM jobs from the VIM (VNFM_PM_VR_PMJOB_DELETE)</li> <li>VIM supports VR PM jobs deletion by the VNFM (VIM_PM_PMJOB_DELETE_BY_VNFM)</li> <li>NFVO can delete VNF PM thresholds from the VNFM (NFVO_PM_VNF_PMTH_DELETE)</li> <li>VNFM supports VNF PM thresholds deletion by the NFVO (VNFM_PM_VNF_PMTH_DELETE)</li> <li>VNFM can delete VR PM thresholds from the VIM (VNFM_PM_VR_PMTH_DELETE)</li> <li>VIM supports VR PM thresholds deletion by the VNFM (VIM_PM_PMTH_DELETE_BY_VNFM)</li> <li>NFVO can subscribe to VNF PM jobs on the VNFM (NFVO_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM supports VNF PM job subscriptions from the NFVO (VNFM_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM can subscribe to VR PM jobs on the VIM (VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the VNFM (VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>VNFM supports VR PM job notifications from the VIM (VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>VNFM can generate PM notifications to the NFVO (VNFM_PM_VNF_PMJOB_NOTIFY)</li> <li>NFVO supports VNF PM job notifications from the VNFM (NFVO_PM_VNF_PMJOB_NOTIFY)</li> <li>VIM supports VR PM job queries by the VNFM (VIM_PM_PMJOB_QUERY_BY_VNFM)</li> <li>NFVO can query VNF PM jobs from the VNFM (NFVO_PM_VNF_PMJOB_QUERY)</li> <li>VNFM supports VNF PM job queries by the NFVO (VNFM_PM_VNF_PMJOB_QUERY)</li> <li>VNFM can query VR PM thresholds from the VIM (VNFM_PM_VR_PMTH_QUERY)</li> <li>VIM supports VR PM threshold queries by the VNFM (VIM_PM_PMTH_QUERY_BY_VNFM)</li> <li>NFVO can query VNF PM thresholds from the VNFM (NFVO_PM_VNF_PMTH_QUERY)</li> <li>VNFM supports VNF PM threshold queries by the NFVO (VNFM_PM_VNF_PMTH_QUERY)</li> </ul>			
Pre-test conditions	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>A virtualised resource that is allocated to a VNF instance inside the target NS instance is monitored by the NFVO (TD_NFV_PM_VNF_CREATE_NOTIFY_001)</li> </ul>			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Trigger the NFVO to delete a PM job of a virtualised resource that is allocated to a VNF instance inside the target NS instance	
	2	IOP Check	Verify that the relevant VNF PM job has been deleted on the VNFM by querying the VNF PM jobs	
	3	IOP Check	Verify that the relevant VR PM job has been deleted on the VIM by querying the VR PM jobs	
	4	IOP Check	Verify that no "performance information available" notification for the monitored virtualised resource has been generated by the VIM to the VNFM by monitoring the VR PM notifications. See note.	

Test Description: VNF PM job deletion			
	5	IOP Check	Verify that no "performance information available" notification for the monitored virtualised resource has been generated by the VNFM to the NFVO by monitoring the VNF PM notifications. See note.
<b>IOP Verdict</b>			
NOTE: Virtualised resource metric collection and reporting periods should be taken into account.			

#### 7.6.2.4 VNF PM Threshold Deletion

Test Description: VNF PM threshold deletion				
<b>Identifier</b>	TD_NFV_PM_VNF_DELETE_THRESHOLD_001			
<b>Test Purpose</b>	Verify that a threshold created for a virtualised resource that is allocated to a VNF instance inside a NS instance can be deleted			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can delete VNF PM thresholds from the VNFM (NFVO_PM_VNF_PMTH_DELETE)</li> <li>VNFM supports VNF PM thresholds deletion by the NFVO (VNFM_PM_VNF_PMTH_DELETE)</li> <li>VNFM can delete VR PM thresholds from the VIM (VNFM_PM_VR_PMTH_DELETE)</li> <li>VIM supports VR PM thresholds deletion by the VNFM (VIM_PM_PMTH_DELETE_BY_VNFM)</li> <li>NFVO can subscribe to VNF PM jobs on the VNFM (NFVO_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM supports VNF PM job subscriptions from the NFVO (VNFM_PM_VNF_PMJOB_SUBSCRIBE)</li> <li>VNFM can subscribe to VR PM jobs on the VIM (VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the VNFM (VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>VNFM supports VR PM job notifications from the VIM (VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>VNFM can query VR PM jobs from the VIM (VNFM_PM_VR_PMJOB_QUERY)</li> <li>VIM supports VR PM job queries by the VNFM (VIM_PM_PMJOB_QUERY_BY_VNFM)</li> <li>NFVO can query VNF PM jobs from the VNFM (NFVO_PM_VNF_PMJOB_QUERY)</li> <li>VNFM supports VNF PM job queries by the NFVO (VNFM_PM_VNF_PMJOB_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>A threshold for a virtualised resource that is allocated to a VNF instance inside a NS instance is created by the NFVO (TD_NFV_PM_VNF_CREATE_THRESHOLD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to delete a VNF PM threshold of a virtualised resource that is allocated to a VNF instance inside the target NS instance	
	2	IOP Check	Verify that the relevant VNF PM threshold has been deleted on the VNFM by querying the VNF PM thresholds	
	3	IOP Check	Verify that the relevant VR PM threshold has been deleted on the VIM by querying the VR PM thresholds	
	4	Stimulus	Trigger the virtualised resource to cross the specified threshold (e.g. by increasing resource utilization levels in the virtualisation container)	
	5	IOP Check	Verify that no "threshold crossed" notification for the monitored virtualised resource has been generated by the VNFM to the NFVO by monitoring the VNF PM notifications. See note.	
	6	IOP Check	Verify that no "threshold crossed" notification for the monitored virtualised resource has been generated by the VIM to the VNFM by monitoring the VR PM notifications. See note.	
<b>IOP Verdict</b>				
NOTE: Virtualised resource metric collection and reporting periods should be taken into account.				

## 7.7 NS Lifecycle Management

### 7.7.1 NS Instantiation

#### 7.7.1.1 Standalone NS Instantiation

Test Description: standalone NS instantiation				
<b>Identifier</b>	TD_NFV_NSLCM_INSTANTIATE_001			
<b>Test Purpose</b>	To verify that a standalone NS can be successfully instantiated			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>• NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>• NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>• VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>• VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>• VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>• NFVO/VNFM can query software image information from the VIM (NFVO_SWIM_QUERY_IM or VNFM_SWIM_QUERY_IM)</li> <li>• VIM supports software image information queries by the NFVO/VNFM (VIM_SWIM_QUERY_IM_BY_NFVO or VIM_SWIM_QUERY_IM_BY_VNFM)</li> <li>• NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>• NFVO/VNFM can query network resource information from the VIM (NFVO_NRM_QUERY or VNFM_NRM_QUERY)</li> <li>• NFVO/VNFM can query storage resource information from the VIM (NFVO_SRM_QUERY or VNFM_SRM_QUERY)</li> <li>• VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>• VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>• VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>• NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>• VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NSD, its associated descriptors (VLD(s), VNFFGD(s)) and VNF Package(s) have been on-boarded to the NFVO</li> <li>• The software image repository is reachable by the VIM</li> <li>• The required resources are available on the NFVI</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS instantiation on the NFVO	
	2	IOP Check	Verify that the software images have been successfully added to the image repository managed by the VIM	
	3	IOP Check	Verify that the requested resources have been allocated by the VIM according to the descriptors	
	4	IOP Check	Verify that the VNF instance(s) have been deployed according to the NSD (i.e. query the VIM and VNFM for VMs, VLs and CPs)	
	5	IOP Check	Verify that the VNF instance(s) are reachable via the management network	
	6	IOP Check	Verify that the VNF instance(s) have been configured according to the VNFD(s) by querying the VNFM	

	7	IOP Check	Verify that the VNF instance(s), VL(s) and VNFFG(s) have been connected according to the descriptors	
	8	IOP Check	Verify that the NFVO indicates NS instantiation operation result as successful	
	9	IOP Check	Verify that the NS is successfully instantiated by running the end-to-end functional test	
<b>IOP Verdict</b>				

### 7.7.1.2 Nested NS Instantiation

<b>Test Description: nested NS instantiation</b>				
<b>Identifier</b>	TD_NFV_NSLCM_INSTANTIATE_NEST_NS_001			
<b>Test Purpose</b>	To verify that a NS referencing an existing nested NS can be successfully instantiated			
<b>Configuration</b>	SUT Configuration 3			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can query software image information from the VIM (NFVO_SWIM_QUERY_IM or VNFM_SWIM_QUERY_IM)</li> <li>VIM supports software image information queries by the NFVO/VNFM (VIM_SWIM_QUERY_IM_BY_NFVO or VIM_SWIM_QUERY_IM_BY_VNFM)</li> <li>VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NSD2, its associated descriptors (VLD(s), VNFFGD(s)) and VNF Package(s) have been on-boarded to the NFVO</li> <li>NSD2 references nested NSD1</li> <li>NS2 is not instantiated</li> <li>NS1 is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The software image repository is reachable by the VIM</li> <li>The required resources are available on the NFVI</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS2 instantiation on the NFVO	
	2	IOP Check	Verify that the software images of the VNF(s) referenced in NSD2 have been successfully added to the image repository managed by the VIM	
	3	IOP Check	Verify that resources associated to NS2 have been allocated by the VIM according to the descriptors	
	4	IOP Check	Verify that the VNF instance(s) have been deployed according to the NSD (i.e. query the VIM and VNFM for VMs, VLs and CPs)	
	5	IOP Check	Verify that existing VNF instance(s) in NS1 are running and reachable via the management network	
	6	IOP Check	Verify that the VNF instance(s) in NS2 are running and reachable through the management network	

Test Description: nested NS instantiation				
	7	IOP Check	Verify that the VNF instances(s) in NS2 have been configured according to the VNFD(s) by querying the VNFM	
	8	IOP Check	Verify that the VNF instance(s), VL(s) and VNFFG(s) in NS2 have been connected according to the descriptors	
	9	IOP Check	Verify that the NFVO indicates NS2 instantiation operation result as successful	
	10	IOP Check	Verify that NS2 is successfully instantiated by running an end-to-end functional test re-using the functionality of VNF instance(s) inside NS1	
<b>IOP Verdict</b>				

## 7.7.2 NS Scaling

### 7.7.2.1 NS Scale Out

#### 7.7.2.1.1 NS Scale out with an Operator Action

Test Description: NS scale out with an operator action				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_OUT_001			
<b>Test Purpose</b>	Verify that the NS can be successfully scaled out by adding VNF instances triggered by an operator action			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO can generate "scale out by adding VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_OUT)</li> <li>VNFM supports "scale out by adding VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_OUT)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out NS</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS scale out by adding VNF instances to the NS in NFVO with an operator action	
	2	IOP Check	Verify that the additional VNF instance(s) have been deployed by querying the VNFM	
	3	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	4	IOP Check	Verify that the additional VNF instance(s) are running and reachable via their management network	
	5	IOP Check	Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors	
	7	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	8	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity	

Test Description: NS scale out with an operator action	
IOP Verdict	

## 7.7.2.1.2 NS Scale out with a VNF Indicator

Test Description: NS scale out with a VNF indicator																																									
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_OUT_002																																								
<b>Test Purpose</b>	Verify that the NS can be successfully scaled out by adding VNF instances triggered automatically by a VNF indicator																																								
<b>Configuration</b>	SUT Configuration 1																																								
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]																																								
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>VNFM can subscribe to VNF indicators on the EM/VNF (VNFM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF supports VNF indicator subscriptions from the VNFM (EM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF can generate VNF indicator notifications to the VNFM (EM_VNFINDI_NOTIFY)</li> <li>VNFM supports VNF indicator notifications from the EM/VNF (VNFM_VNFINDI_NOTIFY)</li> <li>NFVO can generate "scale out by adding VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_OUT)</li> <li>VNFM supports "scale out by adding VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_OUT)</li> </ul>																																								
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out NS</li> <li>NFVO/VNFM is configured to trigger "scale out by adding VNF instances" when a given VNF indicator value crosses a certain threshold</li> </ul>																																								
<b>Test Sequence</b>	<table border="1"> <thead> <tr> <th>Step</th> <th>Type</th> <th>Description</th> <th>Result</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Stimulus</td> <td>Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed</td> <td></td> </tr> <tr> <td>2</td> <td>IOP Check</td> <td>Verify that the "scale out by adding VNF instance(s)" procedure has been started in NFVO</td> <td></td> </tr> <tr> <td>3</td> <td>IOP Check</td> <td>Verify that the additional VNF instance(s) have been deployed by querying the VNFM</td> <td></td> </tr> <tr> <td>4</td> <td>IOP Check</td> <td>Verify that the additional resources have been allocated by 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 via their management network</td> <td></td> </tr> <tr> <td>6</td> <td>IOP Check</td> <td>Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM</td> <td></td> </tr> <tr> <td>7</td> <td>IOP Check</td> <td>Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors</td> <td></td> </tr> <tr> <td>8</td> <td>IOP Check</td> <td>Verify that the NFVO indicates the scaling operation result as successful</td> <td></td> </tr> <tr> <td>9</td> <td>IOP Check</td> <td>Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity</td> <td></td> </tr> </tbody> </table>	Step	Type	Description	Result	1	Stimulus	Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed		2	IOP Check	Verify that the "scale out by adding VNF instance(s)" procedure has been started in NFVO		3	IOP Check	Verify that the additional VNF instance(s) have been deployed by querying the VNFM		4	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors		5	IOP Check	Verify that the additional VNF instance(s) are running and reachable via their management network		6	IOP Check	Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM		7	IOP Check	Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors		8	IOP Check	Verify that the NFVO indicates the scaling operation result as successful		9	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity	
Step	Type	Description	Result																																						
1	Stimulus	Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed																																							
2	IOP Check	Verify that the "scale out by adding VNF instance(s)" procedure has been started in NFVO																																							
3	IOP Check	Verify that the additional VNF instance(s) have been deployed by querying the VNFM																																							
4	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors																																							
5	IOP Check	Verify that the additional VNF instance(s) are running and reachable via their management network																																							
6	IOP Check	Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM																																							
7	IOP Check	Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors																																							
8	IOP Check	Verify that the NFVO indicates the scaling operation result as successful																																							
9	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity																																							
<b>IOP Verdict</b>																																									



## 7.7.2.1.3 NS Scale out with a VIM KPI

Test Description: NS scale out with a VIM KPI				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_OUT_003			
<b>Test Purpose</b>	Verify that the NS can be successfully scaled out by adding VNF instances triggered automatically by a VIM KPI			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_CREATE or VNFM_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the NFVO/VNFM (VIM_PM_PMJOB_CREATE_BY_NFVO or VIM_PM_PMJOB_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM thresholds on the VIM (NFVO_PM_VR_PMTH_CREATE or VNFM_PM_VR_PMTH_CREATE)</li> <li>VIM supports VR PM thresholds creation by the NFVO/VNFM (VIM_PM_PMTH_CREATE_BY_NFVO or VIM_PM_PMTH_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE or VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO or VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the NFVO/VNFM (VIM_PM_PMJOB_NOTIFY_BY_NFVO or VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>NFVO/VNFM supports VR PM notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY or VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can generate "scale out by adding VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_OUT)</li> <li>VNFM supports "scale out by adding VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_OUT)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out NS</li> <li>NFVO/VNFM is configured to trigger "scale out by adding VNF instances" when a given VIM KPI value crosses a certain threshold</li> <li>Virtualised resource PM job with threshold monitoring has been created (TD_NFV_PM_VR_CREATE_THRESHOLD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale out by adding VNF instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the additional VNF instance(s) have been deployed by querying the VNFM	
	4	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNF instance(s) are running and reachable via their management network	
	6	IOP Check	Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM	
	7	IOP Check	Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors	
	8	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	

Test Description: NS scale out with a VIM KPI			
	9	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity
<b>IOP Verdict</b>			

## 7.7.2.2 NS Scale In

### 7.7.2.2.1 NS Scale In with an Operator Action

Test Description: NS scale in with an operator action				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_IN_001			
<b>Test Purpose</b>	Verify that the NS can be successfully scaled in by removing VNF instances triggered by an operator action			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>NFVO can generate "scale in by removing VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_IN)</li> <li>VNFM supports "scale in by removing VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_IN)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NFVO supports triggering "scale in by removing VNF instances" with an operator's action</li> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The current NS deployment size supports scaling in</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS scale in by removing VNF instances from the NS in NFVO with an operator action	
	2	IOP Check	Verify that the impacted VNF instance(s) have been terminated by querying the VNFM	
	3	IOP Check	Verify that the impacted VNF related resources have been released by the VIM	
	4	IOP Check	Verify that the remaining VNF instances(s) are still running and reachable via their management network	
	5	IOP Check	Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors	
	6	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	7	IOP Check	Verify that NS has been scaled in by running the end-to-end functional test factoring the VNF scale and capacity	
<b>IOP Verdict</b>				

## 7.7.2.2.2 NS Scale in with a VNF Indicator

<b>Test Description: NS scale in with a VNF indicator</b>				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_IN_002			
<b>Test Purpose</b>	Verify that the NS can be successfully scaled in by removing VNF instances triggered automatically by a VNF indicator			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>VNFM can subscribe to VNF indicators on the EM/VNF (VNFM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF supports VNF indicator subscriptions from the VNFM (EM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF can generate VNF indicator notifications to the VNFM (EM_VNFINDI_NOTIFY)</li> <li>VNFM supports VNF indicator notifications from the EM/VNF (VNFM_VNFINDI_NOTIFY)</li> <li>NFVO can subscribe to VNF indicators on the VNFM (NFVO_NSVNFINDI_SUBSCRIBE)</li> <li>VNFM supports VNF indicator subscriptions from the NFVO (VNFM_NSVNFINDI_SUBSCRIBE)</li> <li>VNFM can generate VNF indicator notifications to the NFVO (VNFM_NSVNFINDI_NOTIFY)</li> <li>NFVO supports VNF indicator notifications from the VNFM (NFVO_NSVNFINDI_NOTIFY)</li> <li>NFVO can generate "scale in by removing VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_IN)</li> <li>VNFM supports "scale in by removing VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_IN)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NFVO is configured to trigger "scale in by removing VNF instances" when a given VNF indicator value crosses a certain threshold</li> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The current NS deployment size should support scaling in</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale in by removing VNF instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the impacted VNF instance(s) have been terminated by querying the VNFM	
	4	IOP Check	Verify that the impacted VNF related resources have been released by the VIM	
	5	IOP Check	Verify that the remaining VNF instance(s) are still running and reachable via their management network	
	6	IOP Check	Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors	
	7	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	8	IOP Check	Verify that NS has been scaled in by running the end-to-end functional test factoring the VNF scale and capacity	
<b>IOP Verdict</b>				

## 7.7.2.2.3 NS Scale in with a VIM KPI

Test Description: NS scale in with a VIM KPI				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_IN_003			
<b>Test Purpose</b>	Verify that the NS can be successfully scaled in by removing VNF instances triggered automatically by a VIM KPI			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_CREATE or VNFM_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the NFVO/VNFM (VIM_PM_PMJOB_CREATE_BY_NFVO or VIM_PM_PMJOB_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM thresholds on the VIM (NFVO_PM_VR_PMTH_CREATE or VNFM_PM_VR_PMTH_CREATE)</li> <li>VIM supports VR PM thresholds creation by the NFVO/VNFM (VIM_PM_PMTH_CREATE_BY_NFVO or VIM_PM_PMTH_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE or VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO or VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the NFVO/VNFM (VIM_PM_PMJOB_NOTIFY_BY_NFVO or VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>NFVO/VNFM supports VR PM notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY or VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can generate "scale in by removing VNF instances" requests to the VNFM (NFVO_VNFLCM_NS_SCALE_IN)</li> <li>VNFM supports "scale in by removing VNF instances" requests from the NFVO (VNFM_VNFLCM_NS_SCALE_IN)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The current NS deployment size should support scaling in</li> <li>NFVO/VNFM is configured to trigger "scale in by removing VNF instances" when a given VIM KPI value crosses a certain threshold</li> <li>Virtualised resource PM job with threshold monitoring has been created (TD_NFV_PM_VR_CREATE_THRESHOLD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale in by removing VNF instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the impacted VNF instance(s) have been terminated by querying the VNFM	
	4	IOP Check	Verify that the impacted VNF related resources have been released by the VIM	
	5	IOP Check	Verify that the remaining VNF instance(s) are still running and reachable via their management network	
	6	IOP Check	Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors	
	7	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	8	IOP Check	Verify that NS has been scaled in by running the end-to-end functional test factoring the VNF scale and capacity	

Test Description: NS scale in with a VIM KPI	
IOP Verdict	

### 7.7.2.3 NS VNF Scale Out

#### 7.7.2.3.1 NS VNF Scale Out with an Operator Action

Test Description: NS VNF scale out with an operator action				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_OUT_VNF_001			
<b>Test Purpose</b>	To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered by a NFVO operator			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO can generate "scale out by adding VNFC instances" to the VNFM (NFVO_VNFLCM_VNF_SCALE_OUT)</li> <li>VNFM supports "scale out by adding VNFC instances" requests from the NFVO (VNFM_VNFLCM_VNF_SCALE_OUT)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NFVO supports triggering "scale out by adding VNFC instances" with an operator's action</li> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out NS</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS scale out by adding VNFC instance(s) to a VNF in the NS in NFVO with an operator action	
	2	IOP Check	Verify that the additional VNFC instance(s) have been deployed for the VNF by querying the VNFM	
	3	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	4	IOP Check	Verify that the additional VNFC instance(s) are running and reachable via the management network	
	5	IOP Check	Verify that the VNF configuration has been updated to include the additional VNFC instances according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors	
	7	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	8	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test in relevance to the VNF scale and capacity	
<b>IOP Verdict</b>				

## 7.7.2.3.2 NS VNF Scale Out with a VNF Indicator

Test Description: NS VNF scale out with a VNF indicator				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_OUT_VNF_002			
<b>Test Purpose</b>	To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered automatically by a VNF indicator			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>VNFM can subscribe to VNF indicators on the EM/VNF (VNFM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF supports VNF indicator subscriptions from the VNFM (EM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF can generate VNF indicator notifications to the VNFM (EM_VNFINDI_NOTIFY)</li> <li>VNFM supports VNF indicator notifications from the EM/VNF (VNFM_VNFINDI_NOTIFY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out NS</li> <li>VNFM is configured to trigger "scale out by adding VNFC instances" when a given VNF indicator value crosses a certain threshold</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale out by adding VNFC instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the additional VNFC instance(s) have been deployed by querying the VNFM	
	4	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNFC instance(s) are running and are reachable through their management network	
	6	IOP Check	Verify that the VNF configuration has been updated to include the additional VNFC instances according to the descriptors by querying the VNFM	
	7	IOP Check	Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors	
	8	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	9	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test in relevance to the VNF scale and capacity	
<b>IOP Verdict</b>				

## 7.7.2.3.3 NS VNF Scale Out with a VIM KPI

Test Description: NS VNF scale out with a VIM KPI				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_OUT_VNF_003			
<b>Test Purpose</b>	To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered automatically by a VIM KPI			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_CREATE or VNFM_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the NFVO/VNFM (VIM_PM_PMJOB_CREATE_BY_NFVO or VIM_PM_PMJOB_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM thresholds on the VIM (NFVO_PM_VR_PMTH_CREATE or VNFM_PM_VR_PMTH_CREATE)</li> <li>VIM supports VR PM thresholds creation by the NFVO/VNFM (VIM_PM_PMTH_CREATE_BY_NFVO or VIM_PM_PMTH_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE or VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO or VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the NFVO/VNFM (VIM_PM_PMJOB_NOTIFY_BY_NFVO or VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>NFVO/VNFM supports VR PM notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY or VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can generate "scale out by adding VNFC instances" to the VNFM (NFVO_VNFLCM_VNF_SCALE_OUT)</li> <li>VNFM supports "scale out by adding VNFC instances" requests from the NFVO (VNFM_VNFLCM_VNF_SCALE_OUT)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NFVI has the required amount of consumable virtual resources to run the scaled-out NS</li> <li>NFVO/VNFM is configured to trigger "scale out by adding VNFC instances" when a given VIM KPI value crosses a certain threshold</li> <li>Virtualised resource PM job with threshold monitoring has been created (TD_NFV_PM_VR_CREATE_THRESHOLD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale out by adding VNFC instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the additional VNFC instance(s) have been deployed by querying the VNFM	
	4	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	5	IOP Check	Verify that the additional VNFC instance(s) are running and are reachable through their management network	
	6	IOP Check	Verify that the VNF configuration has been updated to include the additional VNFC instances according to the descriptors by querying the VNFM	
	7	IOP Check	Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors	
	8	IOP Check	Verify that the NFVO indicates the scaling operation result as	

Test Description: NS VNF scale out with a VIM KPI			
			successful
	9	IOP Check	Verify that NS has been scaled out by running the end-to-end functional test in relevance to the VNF scale and capacity
<b>IOP Verdict</b>			

## 7.7.2.4 NS VNF Scale In

### 7.7.2.4.1 NS VNF Scale In with an Operator Action

Test Description: NS VNF scale in with an operator action				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_IN_VNF_001			
<b>Test Purpose</b>	Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances from an existing VNF triggered by an operator action			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>NFVO can generate "scale in by removing VNFC instances" to the VNFM (NFVO_VNFLCM_VNF_SCALE_IN)</li> <li>VNFM supports "scale in by removing VNFC instances" requests from the NFVO (VNFM_VNFLCM_VNF_SCALE_IN)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NFVO supports triggering scale in with an operator's action</li> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The current NS deployment size should support scaling in</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS scale in by removing VNFC instance(s) from a VNF in the NS in NFVO with an operator action	
	2	IOP Check	Verify that the impacted VNFC instance(s) inside the VNF have been terminated by querying the VNFM	
	3	IOP Check	Verify that the impacted VNFC instance(s) resources have been released by the VIM	
	4	IOP Check	Verify that the remaining VNFC instance(s) are still running and reachable via their management network	
	5	IOP Check	Verify that the VNF configuration has been updated to exclude the removed VNFC instances according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that the remaining VNFC instances(s) and VL(s) are still connected according to the descriptors	
	7	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	8	IOP Check	Verify that NS has been scaled in by running the end-to-end functional test in relevance to the VNF scale and capacity	
<b>IOP Verdict</b>				



## 7.7.2.4.2 NS VNF Scale In with a VNF Indicator

Test Description: NS VNF scale in with a VNF indicator				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_IN_VNF_002			
<b>Test Purpose</b>	Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances triggered automatically by a VNF indicator			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>VNFM can subscribe to VNF indicators on the EM/VNF (VNFM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF supports VNF indicator subscriptions from the VNFM (EM_VNFINDI_SUBSCRIBE)</li> <li>EM/VNF can generate VNF indicator notifications to the VNFM (EM_VNFINDI_NOTIFY)</li> <li>VNFM supports VNF indicator notifications from the EM/VNF (VNFM_VNFINDI_NOTIFY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The current NS deployment size should support scaling in</li> <li>VNFM is configured to trigger "scale in by removing VNFC instances" when a given VNF indicator value crosses a certain threshold</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale in by removing VNFC instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the impacted VNFC instance(s) inside the VNF have been terminated by querying the VNFM	
	4	IOP Check	Verify that the impacted VNFC instance(s) resources have been released by the VIM	
	5	IOP Check	Verify that the remaining VNFC instance(s) are still running and reachable via their management network	
	6	IOP Check	Verify that the VNF configuration has been updated to exclude the removed VNFC instances according to the descriptors by querying the VNFM	
	7	IOP Check	Verify that the remaining VNFC instances(s) and VL(s) are still connected according to the descriptors	
	8	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	9	IOP Check	Verify that NS has been scaled in by running the end-to-end functional test in relevance to the VNF scale and capacity	
<b>IOP Verdict</b>				

## 7.7.2.4.3 NS VNF Scale In with a VIM KPI

Test Description: NS scale in with a VIM KPI				
<b>Identifier</b>	TD_NFV_NSLCM_SCALE_IN_VNF_003			
<b>Test Purpose</b>	Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances triggered automatically by a VIM KPI			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_CREATE or VNFM_PM_VR_PMJOB_CREATE)</li> <li>VIM supports VR PM jobs creation by the NFVO/VNFM (VIM_PM_PMJOB_CREATE_BY_NFVO or VIM_PM_PMJOB_CREATE_BY_VNFM)</li> <li>NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO_PM_VR_PMJOB_SUBSCRIBE or VNFM_PM_VR_PMJOB_SUBSCRIBE)</li> <li>VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM_PM_PMJOB_SUBSCRIBE_BY_NFVO or VIM_PM_PMJOB_SUBSCRIBE_BY_VNFM)</li> <li>NFVO/VNFM can create VR PM thresholds on the VIM (NFVO_PM_VR_PMTH_CREATE or VNFM_PM_VR_PMTH_CREATE)</li> <li>VIM supports VR PM thresholds creation by the NFVO/VNFM (VIM_PM_PMTH_CREATE_BY_NFVO or VIM_PM_PMTH_CREATE_BY_VNFM)</li> <li>VIM can generate VR PM notifications to the NFVO/VNFM (VIM_PM_PMJOB_NOTIFY_BY_NFVO or VIM_PM_PMJOB_NOTIFY_BY_VNFM)</li> <li>NFVO/VNFM supports VR PM notifications from the VIM (NFVO_PM_VR_PMJOB_NOTIFY or VNFM_PM_VR_PMJOB_NOTIFY)</li> <li>NFVO can generate "scale in by removing VNFC instances" to the VNFM (NFVO_VNFLCM_VNF_SCALE_IN)</li> <li>VNFM supports "scale in by removing VNFC instances" requests from the NFVO (VNFM_VNFLCM_VNF_SCALE_IN)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>The current NS deployment size should support scaling in</li> <li>NFVO/VNFM is configured to trigger "scale in by removing VNFC instances" when a given VIM KPI value crosses a certain threshold</li> <li>Virtualised resource PM job with threshold monitoring has been created (TD_NFV_PM_VR_CREATE_THRESHOLD_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the VIM to send notifications of the targeted KPI to the NFVO/VNFM until the configured threshold is crossed	
	2	IOP Check	Verify that the "scale in by removing VNFC instance(s)" procedure has been started in NFVO	
	3	IOP Check	Verify that the impacted VNFC instance(s) inside the VNF have been terminated by querying the VNFM	
	4	IOP Check	Verify that the impacted VNFC instance(s) resources have been released by the VIM	
	5	IOP Check	Verify that the remaining VNFC instance(s) are still running and reachable via their management network	
	6	IOP Check	Verify that the VNF configuration has been updated to exclude the removed VNFC instances according to the descriptors by querying the VNFM	
	7	IOP Check	Verify that the remaining VNFC instances(s) and VL(s) are still connected according to the descriptors	

	8	IOP Check	Verify that the NFVO indicates the scaling operation result as successful	
	9	IOP Check	Verify that NS has been scaled in by running the end-to-end functional test in relevance to the VNF scale and capacity	
<b>IOP Verdict</b>				

## 7.7.3 NS Update

### 7.7.3.1 Start VNF Instance

Test Description: start VNF instance				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_START_001			
<b>Test Purpose</b>	Verify the capability to start a VNF instance inside a NS instance			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can generate "operate VNF" operation requests to the VNFM (NFVO_VNFLCM_OPERATE)</li> <li>VNFM supports "operate VNF" operation requests from the NFVO (VNFM_VNFLCM_OPERATE)</li> <li>NFVO/VNFM can generate "operate compute resource" operation requests to the VIM (NFVO_CRM_OPERATE or VNFM_CRM_OPERATE)</li> <li>VIM supports "operate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_OPERATE_BY_NFVO or VIM_CRM_OPERATE_BY_VNFM)</li> <li>NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>Target VNF is in a stopped operational state. See note.</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to start the target VNF instance inside the NS instance	
	2	IOP Check	Verify that the compute resources allocated to the VNFC instances inside the target VNF instance have been started by querying the VIM	
	3	IOP Check	Verify that other existing compute resources have not been affected by the performed operation by querying the VIM	
	4	IOP Check	Verify that the VNF instance operational state on the VNFM is indicated as "started"	
	5	IOP Check	Verify that the NFVO shows no "operate VNF" operation errors	
	6	IOP Check	Verify that the NS functionality that utilizes the started VNF instance operates successfully by running the end-to-end functional test	
<b>IOP Verdict</b>				
NOTE: In the state STOPPED, the virtualised container(s), where the VNFC instance(s) of the VNF run, are shut down but not terminated.				

## 7.7.3.2 Stop VNF Instance

Test Description: stop VNF instance				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_STOP_001			
<b>Test Purpose</b>	Verify the capability to stop a VNF instance inside a NS instance			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO can generate "operate VNF" operation requests to the VNFM (NFVO_VNFLCM_OPERATE)</li> <li>VNFM supports "operate VNF" operation requests from the NFVO (VNFM_VNFLCM_OPERATE)</li> <li>NFVO/VNFM can generate "operate compute resource" operation requests to the VIM (NFVO_CRM_OPERATE or VNFM_CRM_OPERATE)</li> <li>VIM supports "operate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_OPERATE_BY_NFVO or VIM_CRM_OPERATE_BY_VNFM)</li> <li>NFVO/VNFM can query compute resource information from the VIM (NFVO_CRM_QUERY or VNFM_CRM_QUERY)</li> <li>VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries by the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the NFVO to stop the target VNF instance inside the NS instance	
	2	IOP Check	Verify that the compute resources allocated to the VNFC instances inside the target VNF instance have been stopped by querying the VIM. See note.	
	3	IOP Check	Verify that other existing compute resources have not been affected by the performed operation by querying the VIM	
	4	IOP Check	Verify that the VNF instance operational state on the VNFM is indicated as "stopped"	
	5	IOP Check	Verify that the NFVO shows no "operate VNF" operation errors	
<b>IOP Verdict</b>				
<b>NOTE:</b>	In the state STOPPED, the virtualised container(s), where the VNFC instance(s) of the VNF run, are shut down but not terminated.			

## 7.7.3.3 Instantiate VNF and Add Instance to NS Instance

<b>Test Description: NS update instantiating VNFs and adding instances</b>				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_INST_ADD_VNF_001			
<b>Test Purpose</b>	To verify that one or more VNFs can be instantiated and the instances added to a running NS instance			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO can generate "create VNF identifier" operation requests to the VNFM (NFVO_VNFLCM_CREATE_VNFID)</li> <li>VNFM supports "create VNF identifier" operation requests from the NFVO (VNFM_VNFLCM_CREATE_VNFID)</li> <li>NFVO can generate "instantiate VNF" operation requests to the VNFM (NFVO_VNFLCM_INSTANTIATE)</li> <li>VNFM supports "instantiate VNF" operation requests from the NFVO (VNFM_VNFLCM_INSTANTIATE)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>VNF Packages whose VNFIDs are referred to in the NSD are on-boarded to the NFVO</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a NS update instantiating VNF(s) and adding the resulting instance(s) to a running NS on the NFVO	
	2	IOP Check	Verify that the additional VNF instance(s) have been deployed by querying the VNFM	
	3	IOP Check	Verify that the additional resources have been allocated by the VIM according to the descriptors	
	4	IOP Check	Verify that the additional VNF instance(s) are running and reachable via their management network	
	5	IOP Check	Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that the NFVO indicates the VNF(s) instantiation and instance(s) addition operation result as successful	
	7	IOP Check	Verify that NS has been updated by running the end-to-end functional test that includes the additional VNF instance(s)	
<b>IOP Verdict</b>				

## 7.7.3.4 Remove VNF Instances from a NS Instance

Test Description: NS update removing VNF instances				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_REM_VNF_001			
<b>Test Purpose</b>	To verify that one or more VNF instances can be removed from a running NS instance			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>NFVO can generate "terminate VNF" operation requests to the VNFM (NFVO_VNFLCM_TERMINATE)</li> <li>VNFM supports "terminate VNF" operation requests from the NFVO (VNFM_VNFLCM_TERMINATE)</li> <li>NFVO can generate "delete VNF identifier" operation requests to the VNFM (NFVO_VNFLCM_DELETE_VNFID)</li> <li>VNFM supports "delete VNF identifier" operation requests from the NFVO (VNFM_VNFLCM_DELETE_VNFID)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NS can function without the impacted VNF instance(s)</li> <li>Functional verification of the VNF instance(s) removal is possible (i.e. reduced capacity)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a NS update removing one or more VNF instances from a running NS instance on the NFVO	
	2	IOP Check	Verify that the impacted resources have been terminated by the VIM according to the descriptors	
	3	IOP Check	Verify that the remaining VNF instance(s) are running and reachable via their management network	
	4	IOP Check	NFVO indicates the update operation was successful	
	5	IOP Check	Verify that NS instance has been updated by running the end-to-end functional test factoring the removal of the VNF instance(s)	
<b>IOP Verdict</b>				

## 7.7.3.5 Add Shared VNF Instances to NS Instance

<b>Test Description: NS update adding shared VNF instances</b>				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_ADD_SHVNF_001			
<b>Test Purpose</b>	To verify that one or more shared VNF instances can be added to a running NS instance			
<b>Configuration</b>	SUT Configuration 2			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS1 is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NS1 can utilize shared VNF instances</li> <li>Functional verification of the additional shared VNF instances is possible (i.e. traffic load sharing)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a NS update adding one or more shared VNF instances to NS1 on the NFVO	
	2	IOP Check	Verify that any additional resources associated to NS1 have been allocated and deployed by the VIM according to the updated descriptors	
	3	IOP Check	Verify that the existing VNF instance(s) in NS1 are running and reachable through the management network	
	4	IOP Check	Verify that the additional shared VNF instances(s) have been configured according to the descriptors by querying the VNFM	
	5	IOP Check	Verify that the NFVO indicates the shared VNF instance(s) addition operation result as successful	
	6	IOP Check	Verify that NS1 has been updated by running the end-to-end functional test that includes the additional shared VNF instance(s)	
	7	IOP Check	Verify that NS2 instance was unaffected by the NS update operation by running the relevant end-to-end functional tests that include the shared VNF instance(s)	
<b>IOP Verdict</b>				

## 7.7.3.6 Remove Shared VNF Instances from NS Instance

<b>Test Description: NS update removing shared VNF instances</b>				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_REM_SHVNF_001			
<b>Test Purpose</b>	To verify that one or more shared VNF instances can be removed from a running NS instance			
<b>Configuration</b>	SUT Configuration 2			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS1 is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NS1 can function without the impacted VNF instances</li> <li>Functional verification of the VNF instance(s) removal is possible (i.e. reduced capacity)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a NS update removing one or more existing VNF instances from NS1 on the NFVO	
	2	IOP Check	Verify that any additional resources associated to NS1 have been removed by the VIM according to the updated descriptors	
	3	IOP Check	Verify that the existing VNF instance(s) in NS1 are still running and reachable through the management network	
	4	IOP Check	Verify that the previously shared VNF instances(s) have been configured according to the updated descriptors by querying the VNFM	
	5	IOP Check	Verify that the NFVO indicates the shared VNF instance(s) removal operation result as successful	
	6	IOP Check	Verify that NS1 has been updated by running the end-to-end functional test factoring the shared VNF instance(s) removal	
	7	IOP Check	Verify that NS2 instance was unaffected by the NS update operation by running the relevant end-to-end functional tests that include the previously shared VNF instance(s)	
<b>IOP Verdict</b>				



## 7.7.3.7 Change VNF Deployment Flavour

Test Description: NS update changing VNF instances DF				
<b>Identifier</b>	TD_NFV_NSLCM_UPDATE_VNF_DF_001			
<b>Test Purpose</b>	To verify that the deployment flavour of one or more VNF instances in a NS instance can be changed			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NS contains at least one VNF that accepts multiple deployment flavours</li> <li>The new deployment flavour is expected to impact a functional aspect of the VNF (i.e. scale or performance)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger a NS update changing the deployment flavour (DF) of one or more VNF instances in a NS instance on NFVO	
	2	IOP Check	Verify that the virtualised resources have been updated by the VIM according to the new deployment flavour	
	3	IOP Check	Verify that the impacted VNF instance(s) are running and reachable through the management network	
	4	IOP Check	Verify that the NFVO indicates the VNF DF update operation result as successful	
	5	IOP Check	Verify that NS has been updated by running the end-to-end functional test factoring the new VNF DF	
<b>IOP Verdict</b>				

## 7.7.4 NS Healing

### 7.7.4.1 Partial NS Healing with an Operator Action

Test Description: partial NS healing with an operator action				
<b>Identifier</b>	TD_NFV_NSLCM_HEAL_001			
<b>Test Purpose</b>	Verify that VNF instances inside the NS can be successfully healed when partial NS healing (VNF healing) is triggered by an operator action			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>• NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>• NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>• NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>• VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>• VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>• VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>• NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>• NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>• NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>• VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>• VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>• VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>• NFVO can generate VNF healing requests to the VNFM (NFVO_VNFLCM_HEAL)</li> <li>• VNFM supports VNF healing requests from the NFVO (VNFM_VNFLCM_HEAL)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>• NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>• NS is in a failed state (for example a virtualised resource needed by one or more VNF instances inside the NS has been terminated directly on the VIM)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger partial NS healing (VNF healing) operation on the NFVO with an operator action	
	2	IOP Check	Verify that any additional resources required for the healing process have been allocated by the VIM according to the descriptors	
	3	IOP Check	Verify that other VNF instances(s) inside the NS are still running and reachable via their management network	
	4	IOP Check	Verify that healed VNF instance(s) are running and reachable via their management network	
	5	IOP Check	Verify that the healed VNF instances(s) have been configured according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that any failed resources have been terminated and released by the VIM	
	7	IOP Check	Verify that the NFVO indicates the partial NS healing (VNF healing) operation result as successful	
	8	IOP Check	Verify that NS has been successfully healed by running an end-to-end functional test factoring the healed VNF instance(s)	
<b>IOP Verdict</b>				

## 7.7.4.2 Complete NS Healing with an Operator Action

Test Description: complete NS healing with an operator action				
<b>Identifier</b>	TD_NFV_NSLCM_HEAL_002			
<b>Test Purpose</b>	Verify that a NS can be successfully healed when complete NS healing is triggered by an operator action			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "allocate compute resource" operation requests to the VIM (NFVO_CRM_ALLOCATE or VNFM_CRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate network resource" operation requests to the VIM (NFVO_NRM_ALLOCATE or VNFM_NRM_ALLOCATE)</li> <li>NFVO/VNFM can generate "allocate storage resource" operation requests to the VIM (NFVO_SRM_ALLOCATE or VNFM_SRM_ALLOCATE)</li> <li>VIM supports "allocate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_ALLOCATE_BY_NFVO or VIM_CRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate network resource" operation requests from the NFVO/VNFM (VIM_NRM_ALLOCATE_BY_NFVO or VIM_NRM_ALLOCATE_BY_VNFM)</li> <li>VIM supports "allocate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_ALLOCATE_BY_NFVO or VIM_SRM_ALLOCATE_BY_VNFM)</li> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> <li>NS is in a failed state (for example a virtualised resource needed by one or more VNF instances inside the NS has been terminated directly on the VIM)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger complete NS healing operation on the NFVO with an operator action	
	2	IOP Check	Verify that resources allocated to the faulty NS instance have been terminated by the VIM	
	3	IOP Check	Verify that resources required by the new NS instance have been allocated by the VIM	
	4	IOP Check	Verify that VNF instance(s) inside the new NS instance are running and reachable via their management network	
	5	IOP Check	Verify that the VNF instances(s) inside the new NS instance have been configured according to the descriptors by querying the VNFM	
	6	IOP Check	Verify that the NFVO indicates the complete NS healing operation result as successful	
	7	IOP Check	Verify that NS has been successfully healed by running an end-to-end functional test	
<b>IOP Verdict</b>				

## 7.7.5 NS Termination

### 7.7.5.1 Standalone NS Termination

Test Description: standalone NS termination				
<b>Identifier</b>	TD_NFV_NSLCM_TERMINATE_001			
<b>Test Purpose</b>	To verify that a standalone NS instance can be successfully terminated			
<b>Configuration</b>	SUT Configuration 1			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)</li> <li>VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries from the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NS is instantiated (TD_NFV_NSLCM_INSTANTIATE_001)</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger the termination of the NS instance on the NFVO	
	2	IOP Check	Verify that all the VNF instance(s) have been terminated by querying the VNFM.	
	3	IOP Check	Verify that the resources allocated to the NS and VNF instance(s) have been released by the VIM	
	4	IOP Check	Verify that the NFVO indicates NS instance termination operation result as successful	
<b>IOP Verdict</b>				

## 7.7.5.2 Nested NS Termination

Test Description: nested NS termination				
<b>Identifier</b>	TD_NFV_NSLCM_TERMINATE_NESTED_NS_001			
<b>Test Purpose</b>	To verify that a NS instance referencing an existing nested NS can be successfully terminated			
<b>Configuration</b>	SUT Configuration 3			
<b>References</b>	ETSI GS NFV-IFA 005 [i.3], ETSI GS NFV-IFA 006 [i.4], ETSI GS NFV-IFA 007 [i.5], ETSI GS NFV-IFA 008 [i.6], ETSI GS NFV-IFA 010 [i.7] and ETSI GS NFV-IFA 013 [i.9]			
<b>Applicability</b>	<ul style="list-style-type: none"> <li>NFVO/VNFM can generate "terminate compute resource" operation requests to the VIM (NFVO_CRM_TERMINATE or VNFM_CRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate network resource" operation requests to the VIM (NFVO_NRM_TERMINATE or VNFM_NRM_TERMINATE)</li> <li>NFVO/VNFM can generate "terminate storage resource" operation requests to the VIM (NFVO_SRM_TERMINATE or VNFM_SRM_TERMINATE)</li> <li>VIM supports "terminate compute resource" operation requests from the NFVO/VNFM (VIM_CRM_TERMINATE_BY_NFVO or VIM_CRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate network resource" operation requests from the NFVO/VNFM (VIM_NRM_TERMINATE_BY_NFVO or VIM_NRM_TERMINATE_BY_VNFM)</li> <li>VIM supports "terminate storage resource" operation requests from the NFVO/VNFM (VIM_SRM_TERMINATE_BY_NFVO or VIM_SRM_TERMINATE_BY_VNFM)VIM can send compute resource information to the NFVO (VIM_CRM_INFO_TO_NFVO or VIM_CRM_INFO_TO_VNFM)</li> <li>VIM can send network resource information to the NFVO (VIM_NRM_INFO_TO_NFVO or VIM_NRM_INFO_TO_VNFM)</li> <li>VIM can send storage resource information to the NFVO (VIM_SRM_INFO_TO_NFVO or VIM_SRM_INFO_TO_VNFM)</li> <li>NFVO can query VNF information from the VNFM (NFVO_VNFLCM_QUERY)</li> <li>VNFM supports VNF information queries from the NFVO (VNFM_VNFLCM_QUERY)</li> </ul>			
<b>Pre-test conditions</b>	<ul style="list-style-type: none"> <li>NSD2 references nested NSD1</li> <li>NS2 is instantiated (TD_NFV_NSLCM_INSTANTIATE_NEST_NS_001)</li> <li>NS1 instance can function without the impacted NS2 instance</li> </ul>			
<b>Test Sequence</b>	<b>Step</b>	<b>Type</b>	<b>Description</b>	<b>Result</b>
	1	Stimulus	Trigger NS2 instance termination on the NFVO	
	2	IOP Check	Verify that the resources that were allocated to the VNF instance(s) inside NS2 have been released by the VIM	
	3	IOP Check	Verify that VNF instance(s) inside NS1 are still running and reachable through the management network	
	4	IOP Check	Verify that all VNF instance(s) in NS2 have been terminated by querying the VNFM	
	5	IOP Check	Verify that the NFVO indicates NS2 instance termination operation result as successful	
	6	IOP Check	Verify that NS1 instance was unaffected by NS2 instance termination by running an end-to-end functional test factoring in the functionality of VNF instance(s) in NS1	
<b>IOP Verdict</b>				

## Annex A: Technical Report Card Sample

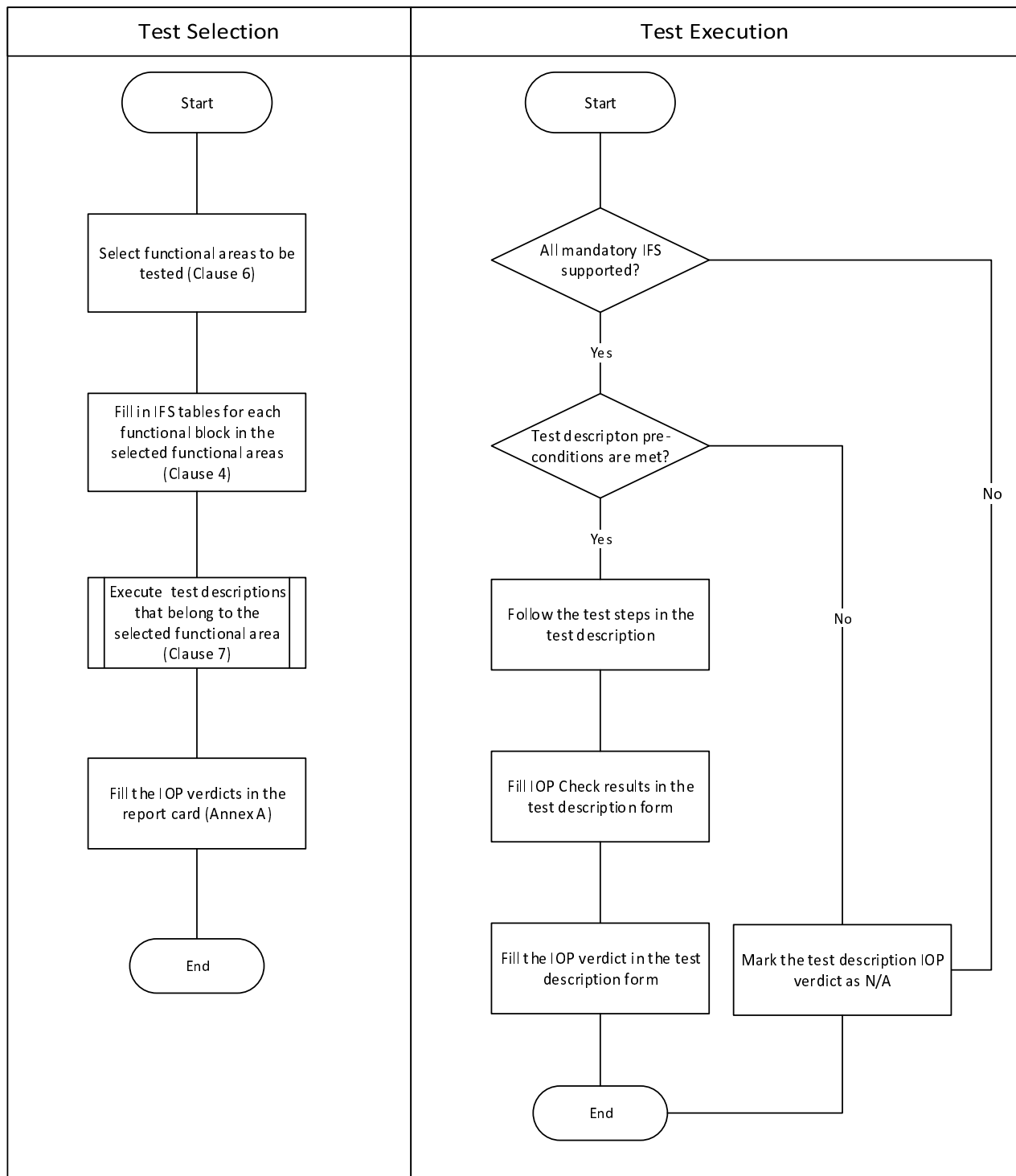
<b>Test ID</b>	xxxx
<b>Tested By</b>	xxxx
<b>Date and Time</b>	dd-mm-yyyy hh:mm

Functional Block	Product Name	Version	Product Description	Configuration
NFVO				
VNFM				
VIM				
NFVI				
EM				
VNF				

Functional Area	Test Description ID	SUT Configuration	IOP Verdict
Software Image Management	TD_NFV_SWIM_ADD_001	SUT Configuration 1	OK
Software Image Management	TD_NFV_SWIM_DELETE_001	SUT Configuration 1	OK
VNF Package Management	TD_NFV_VNFPM_ONBOARD_001	SUT Configuration 1	OK
VNF Package Management	TD_NFV_VNFPM_DELETE_001	SUT Configuration 1	OK
VNF Lifecycle Management	TD_NFV_VNFLCM_INSTANTIATE_VNF_001	SUT Configuration 1	OK
VNF Lifecycle Management	TD_NFV_VNFLCM_QUERY_VNF_001	SUT Configuration 1	OK
VNF Lifecycle Management	TD_NFV_VNFLCM_MODIFY_VNF_INFO_001	SUT Configuration 1	NOK
VNF Lifecycle Management	TD_NFV_VNFLCM_START_VNF_001	SUT Configuration 1	OK
VNF Lifecycle Management	TD_NFV_VNFLCM_STOP_VNF_001	SUT Configuration 1	NOK
VNF Lifecycle Management	TD_NFV_VNFLCM_HEAL_VNF_001	SUT Configuration 1	N/A
VNF Lifecycle Management	TD_NFV_VNFLCM_SCALE_OUT_VNF_001	SUT Configuration 1	NOK
VNF Lifecycle Management	TD_NFV_VNFLCM_SCALE_IN_VNF_001	SUT Configuration 1	NOK
VNF Lifecycle Management	TD_NFV_VNFLCM_TERMINATE_VNF_001	SUT Configuration 1	OK
NS Lifecycle Management	TD_NFV_NSLCM_INSTANTIATE_001	SUT Configuration 1	OK
NS Lifecycle Management	TD_NFV_NSLCM_SCALE_OUT_001	SUT Configuration 1	OK
NS Lifecycle Management	TD_NFV_NSLCM_SCALE_IN_001	SUT Configuration 1	NOK
NS Lifecycle Management	TD_NFV_NSLCM_TERMINATE_001	SUT Configuration 1	OK

## Annex B: Document Usage Process Diagram

This clause explains how the present document can be best utilized in a process diagram.



**Figure B-1: Document usage process diagram**

---

## Annex C: Authors & Contributors

The following people have contributed to the present document:

**Rapporteur:**

Carsten Rossenhoevel, EANTC

**Co-Rapporteur:**

Akram Al Sawaf, EANTC

**Other contributors:**

Al Morton, AT&T

Eduardo Jacob, UPV-EHU

Jokin Garay, UPV-EHU

Jong-Hwa Yi, ETRI

Jörg Aelken, Ericsson

Marie-Paule Odi, Hewlett-Packard Enterprise

Pierre Lynch, Ixia

Silvia Almagia, ETSI Center for Testing and Interoperability (CTI)



---

## History

<b>Document history</b>		
V1.1.1	November 2017	Publication