**Approvals**

|  |  |  |
| --- | --- | --- |
| Prepared by: |  | Reviewed by: |
| <Name>  <Title> |  | <Name>  <Title> |
|  |  |  |
| <Name>  <Title> |  | <Name>  <Title> |
|  |  |  |
| <Name>  <Title> |  | <Name>  <Title> |

# 1 Table of Contents

Table of Contents

[1 Table of Contents 2](#__RefHeading___Toc33348_1326512077)

[2 NS Lifecycle Management Test Cases Overview 4](#__RefHeading___Toc33350_1326512077)

[3 NS Lifecycle Management 10](#__RefHeading___Toc33352_1326512077)

[3.1 NS Instantiation 10](#__RefHeading___Toc492031126)

[3.1.1 Standalone NS Instantiation 10](#__RefHeading___Toc492031127)

[3.1.2 Nested NS Instantiation 12](#__RefHeading___Toc492031128)

[3.2 NS Scaling 13](#__RefHeading___Toc492031129)

[3.2.1 NS Scale Out 13](#__RefHeading___Toc33354_1326512077)

[3.2.1.1 NS Scale out with an Operator Action 13](#__RefHeading___Toc492031131)

[3.2.1.2 NS Scale out with a VNF Indicator 14](#__RefHeading___Toc492031132)

[3.2.1.3 NS Scale out with a VIM KPI 16](#__RefHeading___Toc492031133)

[3.2.2 NS Scale In 17](#__RefHeading___Toc492031134)

[3.2.2.1 NS Scale In with an Operator Action 17](#__RefHeading___Toc492031135)

[3.2.2.2 NS Scale in with a VNF Indicator 18](#__RefHeading___Toc492031136)

[3.2.2.3 NS Scale in with a VIM KPI 20](#__RefHeading___Toc492031137)

[3.2.3 NS VNF Scale Out 21](#__RefHeading___Toc492031138)

[3.2.3.1 NS VNF Scale Out with an Operator Action 21](#__RefHeading___Toc492031139)

[3.2.3.2 NS VNF Scale Out with a VNF Indicator 22](#__RefHeading___Toc492031140)

[3.2.3.3 NS VNF Scale Out with a VIM KPI 24](#__RefHeading___Toc492031141)

[3.2.4 NS VNF Scale In 25](#__RefHeading___Toc492031142)

[3.2.4.1 NS VNF Scale In with an Operator Action 25](#__RefHeading___Toc492031143)

[3.2.4.2 NS VNF Scale In with a VNF Indicator 26](#__RefHeading___Toc492031144)

[3.2.4.3 NS VNF Scale In with a VIM KPI 28](#__RefHeading___Toc492031145)

[3.2.5 NS Scale To Level 29](#__RefHeading___Toc12534_378581709)

[3.2.5.1 NS Scale To Level with an Operator action 29](#__RefHeading___Toc49203114311)

[3.2.5.2 NS Scale To Level with a VNF Indicator 30](#__RefHeading___Toc12495_3785817091)

[3.2.5.3 NS Scale To Level with a VIM KPI 32](#__RefHeading___Toc12497_3785817091)

[3.2.6 NS Scale From Level 33](#__RefHeading___Toc12499_3785817091)

[3.2.6.1 NS Scale From Level with an Operator action 33](#__RefHeading___Toc12501_3785817091)

[3.2.6.2 NS Scale From Level with a VNF Indicator 34](#__RefHeading___Toc12503_3785817091)

[3.2.6.3 NS Scale From Level with a VIM KPI 36](#__RefHeading___Toc49203114511)

[3.2.7 NS VNF Scale To Level 37](#__RefHeading___Toc11366_1322326534)

[3.2.7.1 NS VNF Scale To Level with an Operator action 37](#__RefHeading___Toc4920311431)

[3.2.7.2 NS VNF Scale To Level with a VNF Indicator 38](#__RefHeading___Toc12495_378581709)

[3.2.7.3 NS VNF Scale To Level with a VIM KPI 40](#__RefHeading___Toc12497_378581709)

[3.2.8 NS VNF Scale From Level 41](#__RefHeading___Toc12499_378581709)

[3.2.8.1 NS VNF Scale From Level with an Operator action 41](#__RefHeading___Toc12501_378581709)

[3.2.8.2 NS VNF Scale From Level with a VNF Indicator 43](#__RefHeading___Toc12503_378581709)

[3.2.8.3 NS VNF Scale From Level with a VIM KPI 44](#__RefHeading___Toc4920311451)

[3.3 NS Update 46](#__RefHeading___Toc492031146)

[3.3.1 Start VNF Instance 46](#__RefHeading___Toc492031147)

[3.3.2 Stop VNF Instance 47](#__RefHeading___Toc492031148)

[3.3.3 Instantiate VNF and Add Instance to NS instance 48](#__RefHeading___Toc492031149)

[3.3.4 Remove VNF instances from a NS instance 49](#__RefHeading___Toc492031150)

[3.3.5 Add Shared VNF Instances to NS Instance 50](#__RefHeading___Toc492031151)

[3.3.6 Remove Shared VNF Instances from NS Instance 51](#__RefHeading___Toc492031152)

[3.3.7 Add Virtual Links to a NS Instance 52](#__RefHeading___Toc492031153)

[3.3.8 Remove Virtual Links from a NS Instance 53](#__RefHeading___Toc492031154)

[3.3.9 Change VNF Deployment Flavor 54](#__RefHeading___Toc492031155)

[3.4 NS Healing 55](#__RefHeading___Toc492031156)

[3.4.1 Partial NS Healing with an Operator Action 55](#__RefHeading___Toc492031157)

[3.4.2 Complete NS Healing with an Operator Action 56](#__RefHeading___Toc492031158)

[3.5 NS Termination 58](#__RefHeading___Toc492031159)

[3.5.1 Standalone NS Termination 58](#__RefHeading___Toc492031160)

[3.5.2 Nested NS Termination 59](#__RefHeading___Toc492031161)

[3.6 Multisite 60](#__RefHeading___Toc7937_106121281)

[3.6.1 NS Multisite Instantiation 60](#__RefHeading___Toc7939_106121281)

[3.6.2 NS Multisite Scale Out triggered by MANO operator 61](#__RefHeading___Toc7941_106121281)

[3.6.3 NS Multisite Scale In triggered by MANO operator 62](#__RefHeading___Toc7943_106121281)

[3.6.4 NS Multisite Scale Out triggered by a VNF Indicator 63](#__RefHeading___Toc7945_106121281)

[3.6.5 NS Multisite Scale In triggered by a VNF Indicator 64](#__RefHeading___Toc7947_106121281)

[3.6.6 NS Multisite Scale Out triggered by a VIM KPI 65](#__RefHeading___Toc7949_106121281)

[3.6.7 NS Multisite Scale In triggered by a VIM KPI 66](#__RefHeading___Toc7951_106121281)

[3.6.8 NS Multisite Scale Out triggered by a VNF/EM request 67](#__RefHeading___Toc7953_106121281)

[3.6.9 NS Multisite Scale In triggered by a VNF/EM request 68](#__RefHeading___Toc7955_106121281)

[3.6.10 VNF Scale Out in a multisite NS triggered by a MANO operator 69](#__RefHeading___Toc7957_106121281)

[3.6.11 VNF Scale In in a multisite NS triggered by a MANO operator 70](#__RefHeading___Toc7959_106121281)

[3.7 Enhanced Platform Awareness 71](#__RefHeading___Toc7961_106121281)

[3.7.1 NS EPA Instantiation 71](#__RefHeading___Toc7963_106121281)

[3.7.2 NS with EPA requirements Scale Out triggered by MANO operator 72](#__RefHeading___Toc7965_106121281)

[3.7.3 NS with EPA requirements Scale In triggered by MANO operator 73](#__RefHeading___Toc7967_106121281)

[3.7.4 NS with EPA requirements Scale Out triggered by a VNF Indicator 74](#__RefHeading___Toc7969_106121281)

[3.7.5 NS with EPA requirements Scale In triggered by a VNF Indicator 75](#__RefHeading___Toc7971_106121281)

[3.7.6 NS with EPA requirements Scale Out triggered by a VIM KPI 76](#__RefHeading___Toc7973_106121281)

[3.7.7 NS with EPA requirements Scale In triggered by a VIM KPI 77](#__RefHeading___Toc7975_106121281)

[3.7.8 NS with EPA requirements Scale Out triggered by a VNF/EM request 78](#__RefHeading___Toc7977_106121281)

[3.7.9 NS with EPA requirements Scale In triggered by a VNF/EM request 79](#__RefHeading___Toc7979_106121281)

# 2 NS Lifecycle Management Test Cases Overview

| **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 by a NFVO operator |
| 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\_SCALE\_TO\_LEVEL\_001 | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by an NFVO operator |
| TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_002 | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by a VNF Indicator |
| TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_003 | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_001 | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by an NFVO operator |
| TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_002 | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by a VNF Indicator |
| TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_003 | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_VNF\_001 | Verify that a VNF in a NS can be successfully scaled to an instantiation level when triggered by an NFVO operator |
| TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_VNF\_002 | Verify that a VNF in a NS can be successfully scaled to an instantiation level when triggered by a VNF Indicator |
| TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_VNF\_003 | Verify that a VNF in a NS can be successfully scaled to an instantiation level when triggered by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_VNF\_001 | Verify that a VNF in a NS can be successfully scaled from an instantiation level with a higher number of instances to an instantiation level with a lower number of instances when triggered by an NFVO operator |
| TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_VNF\_002 | Verify that a VNF in a NS can be successfully scaled from an instantiation level with a higher number of instances to an instantiation level with a lower number of instances when triggered by a VNF Indicator |
| TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_VNF\_003 | Verify that a VNF in a NS can be successfully scaled from an instantiation level with a higher number of instances to an instantiation level with a lower number of instances when triggered 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 flavor 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 |
| TD\_NFV\_NSLCM\_INSTANTIATE\_MULTISITE\_001 | To verify that an NS can be successfully instantiated across different sites |
| TD\_NFV\_NSLCM\_INSTANTIATE\_EPA\_001 | To verify that an NS can be successfully instantiated with EPA requirements |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_001 | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered by a MANO operator |
| TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_001 | To verify that a multi-site NS can be successfully scaled in (by adding VNF instances) if triggered by a MANO operator |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_002 | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VNF indicator |
| TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_002 | To verify that a multi-site NS can be successfully scaled in (by adding VNF instances) if triggered automatically in MANO by a VNF indicator |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_003 | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_003 | To verify that a multi-site NS can be successfully scaled in (by adding VNF instances) if triggered automatically in MANO by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_004 | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VNF/EM request |
| TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_004 | To verify that a multi-site NS can be successfully scaled in (by adding VNF instances) if triggered automatically in MANO by a VNF/EM request |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_001 | To verify that an NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically by a MANO operator |
| TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_001 | To verify that an NS can be successfully scaled in with EPA requirements (by adding VNF instances) if triggered automatically by a MANO operator |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_002 | To verify that an NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically in MANO by a VNF indicator |
| TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_002 | To verify that an NS can be successfully scaled in with EPA requirements (by adding VNF instances) if triggered automatically in MANO by a VNF indicator |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_003 | To verify that an NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically in MANO by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_003 | To verify that an NS can be successfully scaled in with EPA requirements (by adding VNF instances) if triggered automatically in MANO by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_004 | To verify that an NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically in MANO by a VNF/EM request |
| TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_004 | To verify that an NS can be successfully scaled in with EPA requirements (by adding VNF instances) if triggered automatically in MANO by a VNF/EM request |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_MULTISITE\_001 | To verify that a multi-site NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered by a MANO operator |
| TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_MULTISITE\_001 | To verify that a multi-site NS can be successfully scaled out (by removing VNFC instances (VMs)) when triggered by a MANO operator |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_MULTISITE\_002 | To verify that a multi-site NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered by a VNF indicator |
| TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_MULTISITE\_002 | To verify that a multi-site NS can be successfully scaled out (by removing VNFC instances (VMs)) when triggered by a VNF indicator |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_MULTISITE\_003 | To verify that a multi-site NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_MULTISITE\_003 | To verify that a multi-site NS can be successfully scaled out (by removing VNFC instances (VMs)) when triggered by a VIM KPI |
| TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_MULTISITE\_004 | To verify that a multi-site NS can be successfully scaled out (by adding VNFC instances (VMs)) when triggered by a VNF/EM request |
| TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_MULTISITE\_004 | To verify that a multi-site NS can be successfully scaled out (by removing VNFC instances (VMs)) when triggered by a VNF/EM request |

# 3 NS Lifecycle Management

## 3.1 NS Instantiation

### 3.1.1 Standalone NS Instantiation

| **Test Description: standalone NS instantiation** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_INSTANTIATE\_001 | | | |
| **Test Purpose** | | To verify that a standalone NS can be successfully instantiated | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can query software image information from the VIM (NFVO\_SWIM\_QUERY\_IM or VNFM\_SWIM\_QUERY\_IM) * VIM supports software image information queries by the NFVO/VNFM (VIM\_SWIM\_QUERY\_IM\_BY\_NFVO or VIM\_SWIM\_QUERY\_IM\_BY\_VNFM) * NFVO/VNFM can query compute resource information from the VIM (NFVO\_CRM\_QUERY or VNFM\_CRM\_QUERY) * NFVO/VNFM can query network resource information from the VIM (NFVO\_NRM\_QUERY or VNFM\_NRM\_QUERY) * NFVO/VNFM can query storage resource information from the VIM (NFVO\_SRM\_QUERY or VNFM\_SRM\_QUERY) * VIM can send compute resource information to the NFVO (VIM\_CRM\_INFO\_TO\_NFVO or VIM\_CRM\_INFO\_TO\_VNFM) * VIM can send network resource information to the NFVO (VIM\_NRM\_INFO\_TO\_NFVO or VIM\_NRM\_INFO\_TO\_VNFM) * VIM can send storage resource information to the NFVO (VIM\_SRM\_INFO\_TO\_NFVO or VIM\_SRM\_INFO\_TO\_VNFM) * NFVO can query VNF information from the VNFM (NFVO\_VNFLCM\_QUERY) * VNFM supports VNF information queries by the NFVO (VNFM\_VNFLCM\_QUERY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NSD, its associated descriptors (VLD(s), VNFFGD(s)) and VNF Package(s) have been on-boarded to the NFVO * The software image repository is reachable by the VIM * The required resources are available on the NFVI | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 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 |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| 11 | | Termination | Calculate NS instantiation time |  |
| **IOP Verdict** |  | | | |  |

### 3.1.2 Nested NS Instantiation

| **Test Description: nested NS instantiation** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_INSTANTIATE\_NEST\_NS\_001 | | | |
| **Test Purpose** | | To verify that a NS referencing an existing nested NS can be successfully instantiated | | | |
| **Configuration** | | SUT Configuration 3 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can query software image information from the VIM (NFVO\_SWIM\_QUERY\_IM or VNFM\_SWIM\_QUERY\_IM) * VIM supports software image information queries by the NFVO/VNFM (VIM\_SWIM\_QUERY\_IM\_BY\_NFVO or VIM\_SWIM\_QUERY\_IM\_BY\_VNFM) * VIM can send compute resource information to the NFVO (VIM\_CRM\_INFO\_TO\_NFVO or VIM\_CRM\_INFO\_TO\_VNFM) * VIM can send network resource information to the NFVO (VIM\_NRM\_INFO\_TO\_NFVO or VIM\_NRM\_INFO\_TO\_VNFM) * VIM can send storage resource information to the NFVO (VIM\_SRM\_INFO\_TO\_NFVO or VIM\_SRM\_INFO\_TO\_VNFM) * NFVO can query VNF information from the VNFM (NFVO\_VNFLCM\_QUERY) * VNFM supports VNF information queries by the NFVO (VNFM\_VNFLCM\_QUERY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports instantiating nested network services * NSD2, its associated descriptors (VLD(s), VNFFGD(s)) and VNF Package(s) have been on-boarded to the NFVO * NSD2 references nested NSD1 * NS2 is not instantiated * NS1 is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The software image repository is reachable by the VIM * The required resources are available on the NFVI | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger nested NS1 instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger NS2 instantiation on the NFVO |  |
| 3 | | 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 |  |
| 4 | | IOP Check | Verify that resources associated to NS2 have been allocated by the VIM according to the descriptors |  |
| 5 | | 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) |  |
| 6 | | IOP Check | Verify that existing VNF instance(s) in NS1 are running and reachable via the management network |  |
| 7 | | IOP Check | Verify that the VNF instance(s) in NS2 are running and reachable through the management network |  |
| 8 | | IOP Check | Verify that the VNF instances(s) in NS2 have been configured according to the VNFD(s) by querying the VNFM |  |
| 9 | | IOP Check | Verify that the VNF instance(s), VL(s) and VNFFG(s) in NS2 have been connected according to the descriptors |  |
| 10 | | IOP Check | Verify that the NFVO indicates NS2 instantiation operation result as successful |  |
| 11 | | 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 |  |
| 12 | | Termination | Trigger NS1 and NS2 termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

## 3.2 NS Scaling

### 3.2.1 NS Scale Out

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

| **Test Description: NS scale out with an operator action** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_001 | | | |
| **Test Purpose** | | Verify that the NS can be successfully scaled out by adding VNF instances triggered by an operator action | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO can generate “scale out by adding VNF instances” requests to the VNFM (NFVO\_VNFLCM\_NS\_SCALE\_OUT) * VNFM supports “scale out by adding VNF instances” requests from the NFVO (VNFM\_VNFLCM\_NS\_SCALE\_OUT) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale out with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled-out NS | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger NS scale out by adding VNF instances to the NS in NFVO with an operator action |  |
| 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 |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.1.2 NS Scale out with a VNF Indicator

| **Test Description: NS scale out with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_002 | | | |
| **Test Purpose** | | Verify that the NS can be successfully scaled out by adding VNF instances triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) * NFVO can generate “scale out by adding VNF instances” requests to the VNFM (NFVO\_VNFLCM\_NS\_SCALE\_OUT) * VNFM supports “scale out by adding VNF instances” requests from the NFVO (VNFM\_VNFLCM\_NS\_SCALE\_OUT) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled-out NS * NFVO/VNFM is configured to trigger “scale out by adding VNF instances” when a given VNF indicator value crosses a certain threshold * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale out by adding VNF instance(s)” procedure has been started in NFVO |  |
| 4 | | IOP Check | Verify that the additional VNF 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 descriptors |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) are running and reachable via their management network |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.1.3 NS Scale out with a VIM KPI

| **Test Description: NS scale out with a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_003 | | | |
| **Test Purpose** | | Verify that the NS can be successfully scaled out by adding VNF instances triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale out by adding VNF instances” requests to the VNFM (NFVO\_VNFLCM\_NS\_SCALE\_OUT) * VNFM supports “scale out by adding VNF instances” requests from the NFVO (VNFM\_VNFLCM\_NS\_SCALE\_OUT) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled-out NS * NFVO/VNFM is configured to trigger “scale out by adding VNF instances” when a given VIM KPI value crosses a certain threshold * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale out by adding VNF instance(s)” procedure has been started in NFVO |  |
| 4 | | IOP Check | Verify that the additional VNF 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 descriptors |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) are running and reachable via their management network |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been scaled out by running the end-to-end functional test factoring the VNF scale and capacity |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.2.2 NS Scale In

#### 3.2.2.1 NS Scale In with an Operator Action

| **Test Description: NS scale in with an operator action** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_001 | | | |
| **Test Purpose** | | Verify that the NS can be successfully scaled in by removing VNF instances triggered by an operator action | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO can generate “scale in by removing VNF instances” requests to the VNFM (NFVO\_VNFLCM\_NS\_SCALE\_IN) * VNFM supports “scale in by removing VNF instances” requests from the NFVO (VNFM\_VNFLCM\_NS\_SCALE\_IN) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering “scale in by removing VNF instances” with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size supports scaling in * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger NS scale in by removing VNF instances from the NS in NFVO with an operator action |  |
| 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 instances(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 |  |
| 9 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.2.2 NS Scale in with a VNF Indicator

| **Test Description: NS scale in with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_002 | | | |
| **Test Purpose** | | Verify that the NS can be successfully scaled in by removing VNF instances triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) * NFVO can subscribe to VNF indicators on the VNFM (NFVO\_NSVNFINDI\_SUBSCRIBE) * VNFM supports VNF indicator subscriptions from the NFVO (VNFM\_NSVNFINDI\_SUBSCRIBE) * VNFM can generate VNF indicator notifications to the NFVO (VNFM\_NSVNFINDI\_NOTIFY) * NFVO supports VNF indicator notifications from the VNFM (NFVO\_NSVNFINDI\_NOTIFY) * NFVO can generate “scale in by removing VNF instances” requests to the VNFM (NFVO\_VNFLCM\_NS\_SCALE\_IN) * VNFM supports “scale in by removing VNF instances” requests from the NFVO (VNFM\_VNFLCM\_NS\_SCALE\_IN) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO is configured to trigger “scale in by removing VNF instances” when a given VNF indicator value crosses a certain threshold * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling in * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale in by removing VNF instance(s)” procedure has been started in NFVO |  |
| 4 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated by querying the VNFM |  |
| 5 | | IOP Check | Verify that the impacted VNF related resources have been released by the VIM |  |
| 6 | | IOP Check | Verify that the remaining VNF instance(s) are still running and reachable via their management network |  |
| 7 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(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 factoring the VNF scale and capacity |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.2.3 NS Scale in with a VIM KPI

| **Test Description: NS scale in with a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_003 | | | |
| **Test Purpose** | | Verify that the NS can be successfully scaled in by removing VNF instances triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale in by removing VNF instances” requests to the VNFM (NFVO\_VNFLCM\_NS\_SCALE\_IN) * VNFM supports “scale in by removing VNF instances” requests from the NFVO (VNFM\_VNFLCM\_NS\_SCALE\_IN) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling in * NFVO/VNFM is configured to trigger “scale in by removing VNF instances” when a given VIM KPI value crosses a certain threshold * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale in by removing VNF instance(s)” procedure has been started in NFVO |  |
| 4 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated by querying the VNFM |  |
| 5 | | IOP Check | Verify that the impacted VNF related resources have been released by the VIM |  |
| 6 | | IOP Check | Verify that the remaining VNF instance(s) are still running and reachable via their management network |  |
| 7 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(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 factoring the VNF scale and capacity |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.2.3 NS VNF Scale Out

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

| **Test Description: NS VNF scale out with an operator action** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_001 | | | |
| **Test Purpose** | | To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered by a NFVO operator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO can generate “scale out by adding VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_OUT) * VNFM supports “scale out by adding VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_OUT) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering “scale out by adding VNFC instances” with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled-out NS * NS/VNF supports scale out by adding VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger NS scale out by adding VNFC instance(s) to a VNF in the NS in NFVO with an operator action |  |
| 3 | | IOP Check | Verify that the additional VNFC instance(s) have been deployed for the VNF 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 reachable via the 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 |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.3.2 NS VNF Scale Out with a VNF Indicator

| **Test Description: NS VNF scale out with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_002 | | | |
| **Test Purpose** | | To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled-out NS * VNFM is configured to trigger “scale out by adding VNFC instances” when a given VNF indicator value crosses a certain threshold * NS/VNF supports scale out by adding VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale out by adding VNFC instance(s)” procedure has been started in 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 descriptors |  |
| 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 descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | 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 |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.3.3 NS VNF Scale Out with a VIM KPI

| **Test Description: NS VNF scale out with a VIM KPI** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_003 | | | | |
| **Test Purpose** | | To verify that a VNF in a NS can be successfully scaled out by adding VNFC instances when triggered automatically by a VIM KPI | | | | |
| **Configuration** | | SUT Configuration 1 | | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale out by adding VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_OUT) * VNFM supports “scale out by adding VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_OUT) | | | | |
|  | | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled-out NS * NFVO/VNFM is configured to trigger “scale out by adding VNFC instances” when a given VIM KPI value crosses a certain threshold * NS/VNF supports scale out by adding VNFC instances | | | | |
|  | | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** | |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  | |
| 2 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  | |
| 3 | | IOP Check | Verify that the “scale out by adding VNFC instance(s)” procedure has been started in 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 descriptors |  | |
| 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 descriptors by querying the VNFM |  | |
| 8 | | IOP Check | Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors |  | |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  | |
| 10 | | 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 |  | |
| 11 | | Termination | Trigger NS termination on NFVO |  | |
| **IOP Verdict** |  | | | | |  |

### 3.2.4 NS VNF Scale In

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

| **Test Description: NS VNF scale in with an operator action** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_001 | | | |
| **Test Purpose** | | 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 | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO can generate “scale in by removing VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_IN) * VNFM supports “scale in by removing VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_IN) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale in with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling in * NS/VNF supports scale in by removing VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger NS scale out by adding VNFC instance(s) to a VNF in the NS in NFVO with an operator action |  |
| 4 | | Stimulus | Trigger NS scale in by removing VNFC instance(s) from a VNF in the NS in NFVO with an operator action |  |
| 5 | | IOP Check | Verify that the impacted VNFC instance(s) inside the VNF have been terminated by querying the VNFM |  |
| 6 | | IOP Check | Verify that the impacted VNFC instance(s) resources have been released by the VIM |  |
| 7 | | IOP Check | Verify that the remaining VNFC instance(s) are still running and reachable via their management network |  |
| 8 | | IOP Check | Verify that the VNF configuration has been updated to exclude the removed VNFC instances according to the descriptors by querying the VNFM |  |
| 9 | | IOP Check | Verify that the remaining VNFC instances(s) and VL(s) are still connected according to the descriptors |  |
| 10 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 11 | | 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 |  |
| 12 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.4.2 NS VNF Scale In with a VNF Indicator

| **Test Description: NS VNF scale in with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_002 | | | |
| **Test Purpose** | | Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling in * VNFM is configured to trigger “scale in by removing VNFC instances” when a given VNF indicator value crosses a certain threshold * NS/VNF supports scale in by removing VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 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 NFVO indicates the scaling operation result as successful |  |
| 10 | | 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 |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.4.3 NS VNF Scale In with a VIM KPI

| **Test Description: NS scale in with a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_003 | | | |
| **Test Purpose** | | Verify that a VNF in a NS can be successfully scaled in by removing VNFC instances triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale in by removing VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_IN) * VNFM supports “scale in by removing VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_IN) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling in * NFVO/VNFM is configured to trigger “scale in by removing VNFC instances” when a given VIM KPI value crosses a certain threshold * NS/VNF supports scale in by removing VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the VIM to send notifications of the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  |
| 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 NFVO indicates the scaling operation result as successful |  |
| 10 | | 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 |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.2.5 NS Scale To Level

#### 3.2.5.1 NS Scale To Level with an Operator action

| **Test Description: NS VNF scale to an instantiation level** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_001 | | | |
| **Test Purpose** | | Verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered by an NFVO operator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO can generate “scale to level by changing the number of VNF instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) * VNFM supports “scale to level by changing the number of VNF instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS supports scale to level by changing the number of VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Stimulus | Trigger NS scale by scaling to another existing instantiation level a VNF in the NS in NFVO with an operator action |  |
| 4 | | IOP Check | Verify that the number of VNF instance(s) has changed for the NS by querying the VNFM |  |
| 5 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 6 | | IOP Check | Verify that all VNF instance(s) are running and reachable via the management network |  |
| 7 | | IOP Check | Verify that the NS configuration has been updated according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that all VNF instance(s) are connected to the VL(s) according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.5.2 NS Scale To Level with a VNF Indicator

| **Test Description: NS VNF scale to an instantiation level with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_002 | | | |
| **Test Purpose** | | To verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled NS * VNFM is configured to trigger “scale to level by changing the number of VNF instances” when a given VNF indicator value crosses a certain threshold * NS supports scale to level by changing the number of VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale to level by adding VNF instance(s)” procedure has been started in NFVO |  |
| 4 | | IOP Check | Verify that the number of VNF instance(s) has changed for the NS by querying the VNFM |  |
| 5 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 6 | | IOP Check | Verify that all VNF instance(s) are running and reachable via the management network |  |
| 7 | | IOP Check | Verify that the NS configuration has been updated according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that all VNF instance(s) are connected to the VL(s) according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.5.3 NS Scale To Level with a VIM KPI

| **Test Description: NS VNF scale to an instantiation level with a VIM KPI** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_003 | | | | |
| **Test Purpose** | | To verify that an NS can be successfully scaled to another existing instantiation level by changing the number of VNF instances when triggered automatically by a VIM KPI | | | | |
| **Configuration** | | SUT Configuration 1 | | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale out by adding VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_OUT) * VNFM supports “scale out by adding VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_OUT) | | | | |
|  | | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled NS * NFVO/VNFM is configured to trigger “scale to level by adding VNFC instances” when a given VIM KPI value crosses a certain threshold * NS/VNF supports scale to level by adding VNFC instances | | | | |
|  | | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** | |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  | |
| 2 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  | |
| 3 | | IOP Check | Verify that the “scale to level by adding VNFC instance(s)” procedure has been started in NFVO |  | |
| 4 | | IOP Check | Verify that the number of VNF instance(s) has changed for the NS by querying the VNFM |  | |
| 5 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  | |
| 6 | | IOP Check | Verify that all VNF instance(s) are running and reachable via the management network |  | |
| 7 | | IOP Check | Verify that the NS configuration has been updated according to the descriptors by querying the VNFM |  | |
| 8 | | IOP Check | Verify that all VNF instance(s) are connected to the VL(s) according to the descriptors |  | |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  | |
| 10 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  | |
| 11 | | Termination | Trigger NS termination on NFVO |  | |
| **IOP Verdict** |  | | | | |  |

### 3.2.6 NS Scale From Level

#### 3.2.6.1 NS Scale From Level with an Operator action

| **Test Description: NS VNF scale from an instantiation level** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_001 | | | |
| **Test Purpose** | | Verify that an NS can be successfully scaled to the initial instantiation level when triggered by an NFVO operator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO can generate “scale to level by changing the number of VNF instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) * VNFM supports “scale to level by changing the number of VNF instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS/VNF supports scale to level by changing the number of VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger NS scale by scaling to another existing instantiation level a VNF in the NS in NFVO with an operator action |  |
| 4 | | Preparation | Verify that the number of VNF instance(s) has changed for the NS by querying the VNFM |  |
| 5 | | Stimulus | Trigger NS scale by scaling to the initial instantiation level in NFVO with an operator action |  |
| 6 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 7 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 8 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 9 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFM |  |
| 10 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 11 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 12 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 13 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.6.2 NS Scale From Level with a VNF Indicator

| **Test Description: NS VNF scale from an instantiation level with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_002 | | | |
| **Test Purpose** | | Verify that an NS can be successfully scaled to the initial instantiation level when triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS supports scale to level by changing the number of VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 4 | | Preparation | Verify that the number of VNF instance(s) has changed for the NS by querying the VNFM |  |
| 5 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed and the NS scales to the initial instantiation level |  |
| 6 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 7 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 8 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 9 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFM |  |
| 10 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 11 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 12 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 13 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.6.3 NS Scale From Level with a VIM KPI

| **Test Description: NS scale from an instantiation level with a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_003 | | | |
| **Test Purpose** | | Verify that an NS can be successfully scaled to the initial instantiation level when triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale to level by changing the number of VNF instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) * VNFM supports “scale to level by changing the number of VNF instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS supports scale to level by changing the number of VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  |
| 4 | | Preparation | Verify that the number of VNF instance(s) has changed for the NS by querying the VNFM |  |
| 5 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed and the NS scales to the initial instantiation level |  |
| 6 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 7 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 8 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 9 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFM |  |
| 10 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 11 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 12 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 13 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.2.7 NS VNF Scale To Level

#### 3.2.7.1 NS VNF Scale To Level with an Operator action

| **Test Description: NS VNF scale to an instantiation level** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_VNF\_001 | | | |
| **Test Purpose** | | Verify that a VNF in a NS can be successfully scaled to another existing instantiation level by changing the number of VNFC instances when triggered by an NFVO operator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO can generate “scale out by adding VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_OUT) * VNFM supports “scale out by adding VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_OUT) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS/VNF supports scale to level by adding/removing VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Stimulus | Trigger NS scale by scaling to another existing instantiation level a VNF in the NS in NFVO with an operator action |  |
| 4 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 5 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 6 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 7 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.7.2 NS VNF Scale To Level with a VNF Indicator

| **Test Description: NS VNF scale to an instantiation level with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_VNF\_002 | | | |
| **Test Purpose** | | To verify that a VNF in a NS can be successfully scaled to another existing instantiation level by changing the number of VNFC instances when triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled NS * VNFM is configured to trigger “scale to level by adding VNFC instances” when a given VNF indicator value crosses a certain threshold * NS/VNF supports scale to level by adding VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the “scale to level by adding VNFC instance(s)” procedure has been started in 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 descriptors |  |
| 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 descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors |  |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been scaled to level by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.7.3 NS VNF Scale To Level with a VIM KPI

| **Test Description: NS VNF scale to an instantiation level with a VIM KPI** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_TO\_LEVEL\_VNF\_003 | | | | |
| **Test Purpose** | | To verify that a VNF in a NS can be successfully scaled to another existing instantiation level by changing the number of VNFC instances when triggered automatically by a VIM KPI | | | | |
| **Configuration** | | SUT Configuration 1 | | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale to level by changing the number of VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_OUT) * VNFM supports “scale to level by changing the number of VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_OUT) | | | | |
|  | | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVI has the required amount of consumable virtual resources to run the scaled NS * NFVO/VNFM is configured to trigger “scale to level by changing the number of VNFC instances” when a given VIM KPI value crosses a certain threshold * NS/VNF supports scale to level by changing the number of VNFC instances | | | | |
|  | | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** | |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  | |
| 2 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  | |
| 3 | | IOP Check | Verify that the “scale to level by adding VNFC instance(s)” procedure has been started in 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 descriptors |  | |
| 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 descriptors by querying the VNFM |  | |
| 8 | | IOP Check | Verify that the additional VNFC instances(s) are connected to the VL(s) according to the descriptors |  | |
| 9 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  | |
| 10 | | 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 |  | |
| 11 | | Termination | Trigger NS termination on NFVO |  | |
| **IOP Verdict** |  | | | | |  |

### 3.2.8 NS VNF Scale From Level

#### 3.2.8.1 NS VNF Scale From Level with an Operator action

| **Test Description: NS VNF scale from an instantiation level** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_VNF\_001 | | | |
| **Test Purpose** | | Verify that a VNF in a NS can be successfully scaled to the initial instantiation level when triggered by an NFVO operator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO can generate “scale to level by changing the number of VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) * VNFM supports “scale to level by changing the number of VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS/VNF supports scale to level by changing the number of VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger NS scale by scaling to another existing instantiation level a VNF in the NS in NFVO with an operator action |  |
| 4 | | Preparation | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 5 | | Stimulus | Trigger NS scale by scaling to the initial instantiation level a VNF in the NS in NFVO with an operator action |  |
| 6 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 7 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 8 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 9 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFM |  |
| 10 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 11 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 12 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 13 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.8.2 NS VNF Scale From Level with a VNF Indicator

| **Test Description: NS VNF scale from an instantiation level with a VNF indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_FROM\_LEVEL\_VNF\_002 | | | |
| **Test Purpose** | | Verify that a VNF in a NS can be successfully scaled to the initial instantiation level when triggered automatically by a VNF indicator | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * VNFM can subscribe to VNF indicators on the EM/VNF (VNFM\_VNFINDI\_SUBSCRIBE) * EM/VNF supports VNF indicator subscriptions from the VNFM (EM\_VNFINDI\_SUBSCRIBE) * EM/VNF can generate VNF indicator notifications to the VNFM (EM\_VNFINDI\_NOTIFY) * VNFM supports VNF indicator notifications from the EM/VNF (VNFM\_VNFINDI\_NOTIFY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS/VNF supports scale to level by changing the number of VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed |  |
| 4 | | Preparation | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 5 | | Stimulus | Trigger the EM/VNF to send the targeted VNF indicator to the VNFM until the configured threshold is crossed and the VNF scales to the initial instantiation level |  |
| 6 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 7 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 8 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 9 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFM |  |
| 10 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 11 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 12 | | IOP Check | Verify that NS has been scaled by running the end-to-end functional test in relevance to the VNF scale and capacity |  |
| 13 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

#### 3.2.8.3 NS VNF Scale From Level with a VIM KPI

| **Test Description: NS scale from an instantiation level with a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_003 | | | |
| **Test Purpose** | | Verify that a VNF in a NS can be successfully scaled to to the initial instantiation level when triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO/VNFM can create VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_CREATE or VNFM\_PM\_VR\_PMJOB\_CREATE) * VIM supports VR PM jobs creation by the NFVO/VNFM (VIM\_PM\_PMJOB\_CREATE\_BY\_NFVO or VIM\_PM\_PMJOB\_CREATE\_BY\_VNFM) * NFVO/VNFM can subscribe to VR PM jobs on the VIM (NFVO\_PM\_VR\_PMJOB\_SUBSCRIBE or VNFM\_PM\_VR\_PMJOB\_SUBSCRIBE) * VIM supports VR PM job subscriptions from the NFVO/VNFM (VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_NFVO or VIM\_PM\_PMJOB\_SUBSCRIBE\_BY\_VNFM) * VIM can generate VR PM notifications to the NFVO/VNFM (VIM\_PM\_PMJOB\_NOTIFY\_BY\_NFVO or VIM\_PM\_PMJOB\_NOTIFY\_BY\_VNFM) * NFVO/VNFM supports VR PM notifications from the VIM (NFVO\_PM\_VR\_PMJOB\_NOTIFY or VNFM\_PM\_VR\_PMJOB\_NOTIFY) * NFVO can generate “scale to level by changing the number of VNFC instances” to the VNFM (NFVO\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) * VNFM supports “scale in by changing the number of VNFC instances” requests from the NFVO (VNFM\_VNFLCM\_VNF\_SCALE\_TO\_LEVEL) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering scale to level with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * The NS initial deployment size should support scaling to a specified level * NS/VNF supports scale to level by changing the number of VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NFVO indicates NS instantiation operation result as successful |  |
| 3 | | Preparation | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed |  |
| 4 | | Preparation | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 5 | | Stimulus | Trigger the VIM to send the targeted KPI to the NFVO/VNFM until the configured threshold is crossed and the VNF scales to the initial instantiation level |  |
| 6 | | IOP Check | Verify that the impacted VNFC instance(s) resources have been released by the VIM |  |
| 7 | | IOP Check | Verify that the number of VNFC instance(s) has changed for the VNF by querying the VNFM |  |
| 8 | | IOP Check | Verify that the resources allocated by the VIM have changed according to the descriptors |  |
| 9 | | IOP Check | Verify that all VNFC instance(s) are running and reachable via the management network |  |
| 10 | | IOP Check | Verify that the VNF configuration has been updated according to the descriptors by querying the VNFMVerify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 11 | | IOP Check | Verify that all VNFC instance(s) are connected to the VL(s) according to the descriptors |  |
| 12 | | IOP Check | Verify that the NFVO indicates the scaling operation result as successful |  |
| 13 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

## 3.3 NS Update

### 3.3.1 Start VNF Instance

| **Test Description: start VNF instance** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_START\_001 | | | |
| **Test Purpose** | | Verify the capability to start a VNF instance inside a NS instance | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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 013 [i.9] | | | |
| **Applicability** | | * NFVO can generate “operate VNF” operation requests to the VNFM (NFVO\_VNFLCM\_OPERATE) * VNFM supports “operate VNF” operation requests from the NFVO (VNFM\_VNFLCM\_OPERATE) * NFVO/VNFM can generate “operate compute resource” operation requests to the VIM (NFVO\_CRM\_OPERATE or VNFM\_CRM\_OPERATE) * VIM supports “operate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_OPERATE\_BY\_NFVO or VIM\_CRM\_OPERATE\_BY\_VNFM) * NFVO/VNFM can query compute resource information from the VIM (NFVO\_CRM\_QUERY or VNFM\_CRM\_QUERY) * VIM can send compute resource information to the NFVO (VIM\_CRM\_INFO\_TO\_NFVO or VIM\_CRM\_INFO\_TO\_VNFM) * VIM can send network resource information to the NFVO (VIM\_NRM\_INFO\_TO\_NFVO or VIM\_NRM\_INFO\_TO\_VNFM) * VIM can send storage resource information to the NFVO (VIM\_SRM\_INFO\_TO\_NFVO or VIM\_SRM\_INFO\_TO\_VNFM) * NFVO can query VNF information from the VNFM (NFVO\_VNFLCM\_QUERY) * VNFM supports VNF information queries by the NFVO (VNFM\_VNFLCM\_QUERY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * Target VNF is in a stopped operational state | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Verify that the NS is instantiated |  |
| 3 | | Preparation | Trigger the NFVO to stop the target VNF instance inside the NS |  |
| 4 | | Stimulus | Trigger the NFVO to start the target VNF instance inside the NS instance |  |
| 5 | | IOP Check | Verify that other existing compute resources have not been affected by the performed operation by querying the VIM |  |
| 6 | | IOP Check | Verify that the compute resources allocated to the target VNF instance have been started by querying the VIM |  |
| 7 | | IOP Check | Verify that the VNF instance operational state on the VNFM is indicated as “started” |  |
| 8 | | IOP Check | Verify that the NFVO shows no “operate VNF” operation errors |  |
| 9 | | IOP Check | Verify that the NS functionality that utilizes the started VNF instance operates successfully by running the end-to-end functional test |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.2 Stop VNF Instance

| **Test Description: stop VNF instance** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_STOP\_001 | | | |
| **Test Purpose** | | Verify the capability to stop a VNF instance inside a NS instance | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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 013 [i.9] | | | |
| **Applicability** | | * NFVO can generate “operate VNF” operation requests to the VNFM (NFVO\_VNFLCM\_OPERATE) * VNFM supports “operate VNF” operation requests from the NFVO (VNFM\_VNFLCM\_OPERATE) * NFVO/VNFM can generate “operate compute resource” operation requests to the VIM (NFVO\_CRM\_OPERATE or VNFM\_CRM\_OPERATE) * VIM supports “operate compute resource” operation requests from the NFVO/VNFM NFVO/VNFM (VIM\_CRM\_OPERATE\_BY\_NFVO or VIM\_CRM\_OPERATE\_BY\_VNFM) * NFVO/VNFM can query compute resource information from the VIM (NFVO\_CRM\_QUERY or VNFM\_CRM\_QUERY) * VIM can send compute resource information to the NFVO (VIM\_CRM\_INFO\_TO\_NFVO or VIM\_CRM\_INFO\_TO\_VNFM) * VIM can send network resource information to the NFVO (VIM\_NRM\_INFO\_TO\_NFVO or VIM\_NRM\_INFO\_TO\_VNFM) * VIM can send storage resource information to the NFVO (VIM\_SRM\_INFO\_TO\_NFVO or VIM\_SRM\_INFO\_TO\_VNFM) * NFVO can query VNF information from the VNFM (NFVO\_VNFLCM\_QUERY) * VNFM supports VNF information queries by the NFVO (VNFM\_VNFLCM\_QUERY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger the NFVO to stop the target VNF instance inside the NS instance |  |
| 3 | | IOP Check | Verify that the compute resources allocated to the target VNF instance have been stopped 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 instance operational state on the VNFM is indicated as “stopped” |  |
| 6 | | IOP Check | Verify that the NFVO shows no “operate VNF” operation errors |  |
| 7 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.3 Instantiate VNF and Add Instance to NS instance

| **Test Description: NS update instantiating VNFs and adding instances** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_INST\_ADD\_VNF\_001 | | | |
| **Test Purpose** | | To verify that one or more VNFs can be instantiated and the instances added to a running NS instance | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO can generate “create VNF identifier” operation requests to the VNFM (NFVO\_VNFLCM\_CREATE\_VNFID) * VNFM supports “create VNF identifier” operation requests from the NFVO (VNFM\_VNFLCM\_CREATE\_VNFID) * NFVO can generate “instantiate VNF” operation requests to the VNFM (NFVO\_VNFLCM\_INSTANTIATE) * VNFM supports “instantiate VNF” operation requests from the NFVO (VNFM\_VNFLCM\_INSTANTIATE) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVO supports triggering NS update instantiating and adding VNF instances to NS * NSD is updated with the revised descriptors and associated to the running NS instance * VNF Packages whose VNFDs are referred to in the NSD are on-boarded to the NFVO | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger a NS update instantiating VNF(s) and adding the resulting instance(s) to a running NS on the 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 NFVO indicates the VNF(s) instantiation and instance(s) addition operation result as successful |  |
| 8 | | IOP Check | Verify that NS has been updated by running the end-to-end functional test that includes the additional VNF instance(s) |  |
| 9 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.4 Remove VNF instances from a NS instance

| **Test Description: NS update removing VNF instances** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_REM\_VNF\_001 | | | |
| **Test Purpose** | | To verify that one or more VNF instances can be removed from a running NS instance | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO can generate “terminate VNF” operation requests to the VNFM (NFVO\_VNFLCM\_TERMINATE) * VNFM supports “terminate VNF” operation requests from the NFVO (VNFM\_VNFLCM\_TERMINATE) * NFVO can generate “delete VNF identifier” operation requests to the VNFM (NFVO\_VNFLCM\_DELETE\_VNFID) * VNFM supports “delete VNF identifier” operation requests from the NFVO (VNFM\_VNFLCM\_DELETE\_VNFID) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVO supports triggering NS update removing VNF instances from NS * NS can function without the impacted VNF instance(s) * Functional verification of the VNF instance(s) removal is possible (i.e. reduced capacity) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 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) |  |
| 6 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.5 Add Shared VNF Instances to NS Instance

| **Test Description: NS update adding shared VNF instances** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_ADD\_SHVNF\_001 | | | |
| **Test Purpose** | | To verify that one or more shared VNF instances can be added to a running NS instance | | | |
| **Configuration** | | SUT Configuration 2 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS1 is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVO supports triggering NS update adding shared VNF instances to NS1 * NS1 can utilize shared VNF instances * NSD1 is updated with the revised descriptors and associated to NS1 * Functional verification of the additional shared VNF instances is possible (i.e. traffic load sharing) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger a NS update adding one or more shared VNF instances to NS1 on the NFVO |  |
| 3 | | IOP Check | Verify that any additional resources associated to NS1 have been allocated and deployed by the VIM according to the updated descriptors |  |
| 4 | | IOP Check | Verify that the existing VNF instance(s) in NS1 are running and reachable through the management network |  |
| 5 | | IOP Check | Verify that the additional shared VNF instances(s) have been configured according to the descriptors by querying the VNFM |  |
| 6 | | IOP Check | Verify that the NFVO indicates the shared VNF instance(s) addition operation result as successful |  |
| 7 | | IOP Check | Verify that NS1 has been updated by running the end-to-end functional test that includes the additional shared VNF instance(s) |  |
| 8 | | 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) |  |
| 9 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.6 Remove Shared VNF Instances from NS Instance

| **Test Description: NS update removing shared VNF instances** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_REM\_SHVNF\_001 | | | |
| **Test Purpose** | | To verify that one or more shared VNF instances can be removed from a running NS instance | | | |
| **Configuration** | | SUT Configuration 2 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS1 is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVO supports triggering NS update removing shared VNF instances from NS1 with an operator's action * NS1 can function without the impacted VNF instances * NSD1 is updated with the revised descriptors and associated to the running NS1 * Functional verification of the VNF instance(s) removal is possible (i.e. reduced capacity) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger a NS update removing one or more existing VNF instances from NS1 on the NFVO |  |
| 3 | | IOP Check | Verify that any additional resources associated to NS1 have been removed by the VIM according to the updated descriptors |  |
| 4 | | IOP Check | Verify that the existing VNF instance(s) in NS1 are still running and reachable through the management network |  |
| 5 | | IOP Check | Verify that the previously shared VNF instances(s) have been configured according to the updated descriptors by querying the VNFM |  |
| 6 | | IOP Check | Verify that the NFVO indicates the shared VNF instance(s) removal operation result as successful |  |
| 7 | | IOP Check | Verify that NS1 has been updated by running the end-to-end functional test factoring the shared VNF instance(s) removal |  |
| 8 | | 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) |  |
| 9 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.7 Add Virtual Links to a NS Instance

| **Test Description: NS update adding VLs to NS instance** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_ADD\_VL\_001 | | | |
| **Test Purpose** | | To verify that one or more virtual links (VL) can be added to a running NS instance | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NFVO supports triggering NS update associating a new NSD to NS * NSD is updated with the revised VL descriptors only | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger a NS update associating the NSD with the additional VL(s) to a running NS instance on the NFVO |  |
| 3 | | IOP Check | Verify that the requested network resources have been allocated and deployed by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the VNF instance(s) inside the impacted NS are still running and reachable through the management network |  |
| 5 | | IOP Check | Verify that the NFVO indicates the new NSD association operation result as successful |  |
| 6 | | IOP Check | Verify that NS has not been affected by the update by running the end-to-end functional test |  |
| 7 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.8 Remove Virtual Links from a NS Instance

| **Test Description: NS update removing VLs from NS instance** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_REM\_VL\_001 | | | |
| **Test Purpose** | | To verify that one or more virtual links (VL) can be removed from a running NS instance | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering NS update associating a new NSD to NS * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NSD is updated with the revised VL descriptors only * NS can function without the impacted VL(s) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger a NS update associating the NSD removing the impacted VL(s) to a running NS instance on the NFVO |  |
| 3 | | IOP Check | Verify that the requested network resources have been terminated by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the NFVO indicates the new NSD association operation result as successful |  |
| 5 | | IOP Check | Verify that NS has been updated by running the end-to-end functional test factoring the removal of the VL(s) |  |
| 6 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.3.9 Change VNF Deployment Flavor

| **Test Description: NS update changing VNF instances DF** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_UPDATE\_VNF\_DF\_001 | | | |
| **Test Purpose** | | To verify that the deployment flavor of one or more VNF instances in a NS instance can be changed | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering NS update to change a VNF DF with an operator’s action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * NS contains at least one VNF that accepts multiple deployment flavors * The new deployment flavor is expected to impact a functional aspect of the VNF (i.e. scale or performance) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Stimulus | Trigger a NS update changing the deployment flavour (DF) of one or more VNF instances in a NS instance on NFVO |  |
| 3 | | IOP Check | Verify that the virtualised resources have been updated by the VIM according to the new deployment flavor |  |
| 4 | | IOP Check | Verify that the impacted VNF instance(s) are running and reachable through the management network |  |
| 5 | | IOP Check | Verify that the NFVO indicates the VNF DF update operation result as successful |  |
| 6 | | IOP Check | Verify that NS has been updated by running the end-to-end functional test factoring the new VNF DF |  |
| 7 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

## 3.4 NS Healing

### 3.4.1 Partial NS Healing with an Operator Action

| **Test Description: partial NS healing with an operator action** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_HEAL\_001 | | | |
| **Test Purpose** | | Verify that VNF instances inside the NS can be successfully healed when partial NS healing (VNF healing) is triggered by an operator action | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) * NFVO can generate VNF healing requests to the VNFM (NFVO\_VNFLCM\_HEAL) * VNFM supports VNF healing requests from the NFVO (VNFM\_VNFLCM\_HEAL) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering partial NS healing (VNF healing) with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * 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) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Trigger a NS fail (for example a virtualised resources needed by one or more VNF instances inside the NS is terminated directly on the VIM) |  |
| 3 | | Stimulus | Trigger partial NS healing (VNF healing) operation on the NFVO with an operator action |  |
| 4 | | IOP Check | Verify that any additional resources required for the healing process have been allocated by the VIM according to the descriptors |  |
| 5 | | IOP Check | Verify that other VNF instances(s) inside the NS are still running and reachable via their management network |  |
| 6 | | IOP Check | Verify that healed VNF instance(s) are running and reachable via their management network |  |
| 7 | | IOP Check | Verify that the healed VNF instances(s) have been configured according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that any failed resources have been terminated and released by the VIM |  |
| 9 | | IOP Check | Verify that the NFVO indicates the partial NS healing (VNF healing) operation result as successful |  |
| 10 | | IOP Check | Verify that NS has been successfully healed by running an end-to-end functional test factoring the healed VNF instance(s) |  |
| 11 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

### 3.4.2 Complete NS Healing with an Operator Action

| **Test Description: complete NS healing with an operator action** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_HEAL\_002 | | | |
| **Test Purpose** | | Verify that a NS can be successfully healed when complete NS healing is triggered by an operator action | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “allocate compute resource” operation requests to the VIM (NFVO\_CRM\_ALLOCATE or VNFM\_CRM\_ALLOCATE) * NFVO/VNFM can generate “allocate network resource” operation requests to the VIM (NFVO\_NRM\_ALLOCATE or VNFM\_NRM\_ALLOCATE) * NFVO/VNFM can generate “allocate storage resource” operation requests to the VIM (NFVO\_SRM\_ALLOCATE or VNFM\_SRM\_ALLOCATE) * VIM supports “allocate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_ALLOCATE\_BY\_NFVO or VIM\_CRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_ALLOCATE\_BY\_NFVO or VIM\_NRM\_ALLOCATE\_BY\_VNFM) * VIM supports “allocate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_ALLOCATE\_BY\_NFVO or VIM\_SRM\_ALLOCATE\_BY\_VNFM) * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate storage resource” operation requests from the NFVO/VNFM (VIM\_SRM\_TERMINATE\_BY\_NFVO or VIM\_SRM\_TERMINATE\_BY\_VNFM) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports triggering complete NS healing with an operator's action * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) * 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) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Preparation | Trigger a NS fail (for example a virtualised resources needed by one or more VNF instances inside the NS is terminated directly on the VIM) |  |
| 3 | | Stimulus | Trigger complete NS healing operation on the NFVO with an operator action |  |
| 4 | | IOP Check | Verify that resources allocated to the faulty NS instance have been terminated by the VIM |  |
| 5 | | IOP Check | Verify that resources required by the new NS instance have been allocated by the VIM |  |
| 6 | | IOP Check | Verify that VNF instance(s) inside the new NS instance are running and reachable via their management network |  |
| 7 | | IOP Check | Verify that the VNF instances(s) inside the new NS instance have been configured according to the descriptors by querying the VNFM |  |
| 8 | | IOP Check | Verify that the NFVO indicates the complete NS healing operation result as successful |  |
| 9 | | IOP Check | Verify that NS has been successfully healed by running an end-to-end functional test |  |
| 10 | | Termination | Trigger NS termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

## 3.5 NS Termination

### 3.5.1 Standalone NS Termination

| **Test Description: standalone NS termination** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_TERMINATE\_001 | | | |
| **Test Purpose** | | To verify that a standalone NS instance can be successfully terminated | | | |
| **Configuration** | | SUT Configuration 1 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * 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) * VIM can send network resource information to the NFVO (VIM\_NRM\_INFO\_TO\_NFVO or VIM\_NRM\_INFO\_TO\_VNFM) * VIM can send storage resource information to the NFVO (VIM\_SRM\_INFO\_TO\_NFVO or VIM\_SRM\_INFO\_TO\_VNFM) * NFVO can query VNF information from the VNFM (NFVO\_VNFLCM\_QUERY) * VNFM supports VNF information queries from the NFVO (VNFM\_VNFLCM\_QUERY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_001) | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation on the NFVO |  |
| 2 | | Verification | Verify that the NS is instantiated and started |  |
| 3 | | Stimulus | Trigger the termination of the NS instance on the NFVO |  |
| 4 | | IOP Check | Verify that all the VNF instance(s) have been terminated by querying the VNFM. |  |
| 5 | | IOP Check | Verify that the resources allocated to the NS and VNF instance(s) have been released by the VIM |  |
| 6 | | IOP Check | Verify that the NFVO indicates NS instance termination operation result as successful |  |
| **IOP Verdict** |  | | | |  |

### 3.5.2 Nested NS Termination

| **Test Description: nested NS termination** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_TERMINATE\_NESTED\_NS\_001 | | | |
| **Test Purpose** | | To verify that a NS instance referencing an existing nested NS can be successfully terminated | | | |
| **Configuration** | | SUT Configuration 3 | | | |
| **References** | | 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], ETSI GS NFV-IFA 013 [i.9] | | | |
| **Applicability** | | * NFVO/VNFM can generate “terminate compute resource” operation requests to the VIM (NFVO\_CRM\_TERMINATE or VNFM\_CRM\_TERMINATE) * NFVO/VNFM can generate “terminate network resource” operation requests to the VIM (NFVO\_NRM\_TERMINATE or VNFM\_NRM\_TERMINATE) * NFVO/VNFM can generate “terminate storage resource” operation requests to the VIM (NFVO\_SRM\_TERMINATE or VNFM\_SRM\_TERMINATE) * VIM supports “terminate compute resource” operation requests from the NFVO/VNFM (VIM\_CRM\_TERMINATE\_BY\_NFVO or VIM\_CRM\_TERMINATE\_BY\_VNFM) * VIM supports “terminate network resource” operation requests from the NFVO/VNFM (VIM\_NRM\_TERMINATE\_BY\_NFVO or VIM\_NRM\_TERMINATE\_BY\_VNFM) * 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) * VIM can send network resource information to the NFVO (VIM\_NRM\_INFO\_TO\_NFVO or VIM\_NRM\_INFO\_TO\_VNFM) * VIM can send storage resource information to the NFVO (VIM\_SRM\_INFO\_TO\_NFVO or VIM\_SRM\_INFO\_TO\_VNFM) * NFVO can query VNF information from the VNFM (NFVO\_VNFLCM\_QUERY) * VNFM supports VNF information queries from the NFVO (VNFM\_VNFLCM\_QUERY) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NFVO supports terminating nested network services * NSD2 references nested NSD1 * NS2 is instantiated (TD\_NFV\_NSLCM\_INSTANTIATE\_NEST\_NS\_001) * NS1 instance can function without the impacted NS2 instance | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS1 (nested) instantiation on the NFVO |  |
| 2 | | Preparation | Trigger NS2 instantiation on the NFVO |  |
| 3 | | Stimulus | Trigger NS2 instance termination on the NFVO |  |
| 4 | | IOP Check | Verify that the resources that were allocated to the VNF instance(s) inside NS2 have been released by the VIM |  |
| 5 | | IOP Check | Verify that VNF instance(s) inside NS1 are still running and reachable through the management network |  |
| 6 | | IOP Check | Verify that all VNF instance(s) in NS2 have been terminated by querying the VNFM |  |
| 7 | | IOP Check | Verify that the NFVO indicates NS2 instance termination operation result as successful |  |
| 8 | | 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 |  |
| 9 | | Termination | Trigger NS1 and NS2 termination on NFVO |  |
| **IOP Verdict** |  | | | |  |

## 3.6 Multisite

### 3.6.1 NS Multisite Instantiation

| **Test Description: multi-site NS instantiation** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_INSTANTIATION\_MULTISITE\_001 | | | |
| **Test Purpose** | | To verify that an NS can be successfully instantiated across different sites | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.3)  ETSI GS NFV-IFA 005 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.2.3)  ETSI GS NFV-IFA 008 v2.3.1 (clause 6.2.3)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.3.2)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.2) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to add a SW image * All VIM\_NFVIs supports adding a SW image * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * (If required by NSD) MANO can request all VIM\_NFVIs to create NFP(s) * \* (If required by NSD) All VIM\_NFVIs supports creating NFP(s) | | | |
|  | | | | | |
| **Pre-test conditions** | | * NSD, VLD(s), VNFFGD(s) and VNF Package(s) have been on-boarded in MANO * The software image repository is reachable by the VIMs * The required resources are available on the NFVIs | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Stimulus | Trigger multi-site NS instantiation in MANO |  |
| 2 | | IOP Check | Verify that the software images have been onboarded in the VIMs |  |
| 3 | | IOP Check | Verify that the requested resources have been allocated by the VIMs according to the descriptors |  |
| 4 | | IOP Check | Verify that the VNF (s) have been deployed according to the desciptors (VMs, VLs, CPs...) |  |
| 5 | | IOP Check | Verify that the VL and VNFFG instance(s) have been created according to the descriptors |  |
| 6 | | IOP Check | Verify that the VNF (s) have been deployed according to the multi-site location constraints |  |
| 7 | | IOP Check | Verify that the VNF(s) are running and reachable through the management network |  |
| 8 | | IOP Check | Verify that the VNF(s) have been configured according to VNFD(s) (i.e by obtaining a result from the management interface) |  |
| 9 | | IOP Check | Verify that the VNF(s), VL(s) and VNFFG(s) have been connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that the VNF(s) have multi-site connectivity through the multi-site VL(s) |  |
| 11 | | IOP Check | Verify that the multi-site NS is successfully instantiated by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.2 NS Multisite Scale Out triggered by MANO operator

| **Test Description: multi-site NS scale out triggered by MANO operator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_001 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered by a MANO operator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * MANO supports triggering scale out with an operator's action * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger multi-site NS scale out (by adding VNF instances) in MANO with an operator action |  |
| 3 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the descriptors (VMs, VLs, CPs...) |  |
| 5 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the multi-site location constraints |  |
| 6 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 9 | | IOP Check | Verify that the VNF(s) have multi-site connectivity through the multi-site VL(s) |  |
| 10 | | IOP Check | Verify that multi-site NS has been scaled out by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.3 NS Multisite Scale In triggered by MANO operator

| **Test Description: multi-site NS scale in triggered by MANO operator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_001 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled in (by removing VNF instances) if triggered by a MANO operator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * MANO supports triggering scale in with an operator's action * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated * Multi-site NS has been scaled out by adding VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Preparation | Trigger multi-site NS scale out in MANO (by adding VNF instances) |  |
| 3 | | Stimulus | Trigger multi-site NS scale in (by removing VNFs) in MANO with an operator action |  |
| 4 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated and not running in the correspondant VIM site / instance |  |
| 5 | | IOP Check | Verify that the impacted VNF related resources have been released by the proper VIM site / VIM instance |  |
| 6 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 7 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 8 | | IOP Check | Verify that the remaining VNF instance(s) have still multi-site connectivity through the multi-site VL(s) |  |
| 9 | | IOP Check | Verify that multi-site NS has been scaled in by running the end-to-end functional test |  |
| 10 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.4 NS Multisite Scale Out triggered by a VNF Indicator

| **Test Description: multi-site NS scale out triggered by a VNF Indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_002 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VNF Indicator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * MANO supports receiving VNF indicators from VNF/EM irrispectively of the location/VIM site * VNF/EM can send VNF indicator values to MANO * MANO supports triggering scale out when a given VNF Indicator value crosses a certain treshold * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated * MANO is configured to trigger SCALE OUT (by adding VNF instances) when a given VNF Indicator value crosses a certain threshold | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger the VNF/EM to send the targeted VNF indicator to MANO until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the scale out (by adding VNF(s)) procedure has been started in MANO |  |
| 4 | | IOP Check | Verify that the requested resources have been allocated by the VIM managing the site where the additional VNF(s) are deployed according to the descriptors |  |
| 5 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to descriptors (VMs, VLs, CPs...) |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the multi-site location constraints (as part of auto-scaling logics)? |  |
| 7 | | IOP Check | Verify that the additional VNF(s) are running and reachable through the management network |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to VNFD (i.e by obtaining a result from the management interface) |  |
| 9 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that the addtional VNF instance(s) have multi-site connectivity through the multi-site VL(s) |  |
| 11 | | IOP Check | Verify that multi-site NS has been scaled out by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.5 NS Multisite Scale In triggered by a VNF Indicator

| **Test Description: multi-site NS scale in triggered by a VNF Indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_002 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled in (by removing VNF instances) if triggered automatically in MANO by a VNF Indicator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * MANO supports receiving VNF indicators from VNF/EM irrispectively of the location/VIM site * VNF/EM can send VNF indicator values to MANO * MANO supports triggering scale in when a given VNF Indicator value crosses a certain treshold * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated * Multi-site NS has been scaled out by adding VNF instances * MANO is configured to trigger SCALE IN (by removing VNF instances) when a given VNF Indicator value crosses a certain threshold | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger the VNF/EM to send the targeted VNF indicator to MANO until the configured threshold is crossed |  |
| 4 | | IOP Check | Verify that the scale in (by removing VNF(s)) procedure has been started in MANO |  |
| 5 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated by the VIM managing the correspondent site |  |
| 6 | | IOP Check | Verify that the impacted VNF related resources have been released by the proper VIM site / VIM instance |  |
| 7 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 8 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 9 | | IOP Check | Verify that the remaining VNF instance(s) have still multi-site connectivity through the multi-site VL(s) |  |
| 10 | | IOP Check | Verify that multi-site NS has been scaled in by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.6 NS Multisite Scale Out triggered by a VIM KPI

| **Test Description: multi-site NS scale out triggered by a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_003 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VIM KPI | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVI to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * MANO supports receiving VR related KPIs from VIM * All VIM\_NFVIs can send VR related KPIs to MANO * MANO supports triggering scale out when a given KPI crosses a certain treshold * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated * MANO is configured to trigger SCALE OUT (by adding VNF instances) when a given VIM KPI value crosses a certain threshold" | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger one of the VIMs to send the targeted KPI to MANO until the configured scale-out threshold is crossed |  |
| 3 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the descriptors (VMs, VLs, CPs...) |  |
| 5 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the multi-site location constraints |  |
| 6 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 9 | | IOP Check | Verify that the addtional VNF instance(s) have multi-site connectivity through the multi-site VL(s) |  |
| 10 | | IOP Check | Verify that the addtional VNF instance(s) have multi-site connectivity through the multi-site VL(s) |  |
| 11 | | IOP Check | Verify that multi-site NS has been scaled out by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.7 NS Multisite Scale In triggered by a VIM KPI

| **Test Description: multi-site NS scale out triggered by a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_003 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled in (by removing VNF instances) if triggered automatically in MANO by a VIM KPI | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to terminate virtualised resources * All VIM\_NFVIs supports termintaing virtualised resources * MANO supports receiving VR related KPIs from VIM * VIM\_NFVI can send VR related KPIs to MANO * MANO supports triggering scale in when a given KPI crosses a certain treshold * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated * Multi-site NS has been scaled out by adding VNF instances * MANO is configured to trigger SCALE IN (by removing VNF instances) when a given VIM KPI value crosses a certain threshold | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger one of the VIM to send the targeted KPI to MANO until the configured scale-in threshold is crossed |  |
| 4 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated and not running in the correspondant VIM site / instance |  |
| 5 | | IOP Check | Verify that the impacted VNF related resources have been released by the proper VIM site / VIM instance |  |
| 6 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 7 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 8 | | IOP Check | Verify that the remaining VNF instance(s) have still multi-site connectivity through the multi-site VL(s) |  |
| 9 | | IOP Check | Verify that multi-site NS has been scaled in by running the end-to-end functional test |  |
| 10 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.8 NS Multisite Scale Out triggered by a VNF/EM request

| **Test Description: multi-site NS scale out triggered by a VNF/EM request** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_MULTISITE\_004 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled out (by adding VNF instances) if triggered automatically in MANO by a VNF/EM request | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs support allocating virtualised resources * VNF/EM can send scale out request to MANO * MANO supports triggering multi-site scale out when the scale out request is received from VNF/EM * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| Pre-test conditions | | * Multi-site NS is instantiated | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger the VNF/EM to send a scale out (by adding VNFs) request to MANO |  |
| 3 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the descriptors (VMs, VLs, CPs...) |  |
| 5 | | IOP Check | Verify that the additional VNF instance(s) have been deployed according to the multi-site location constraints |  |
| 6 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 9 | | IOP Check | Verify that the addtional VNF instance(s) have multi-site connectivity through the multi-site VL(s) |  |
| 10 | | IOP Check | Verify that multi-site NS has been scaled out by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.9 NS Multisite Scale In triggered by a VNF/EM request

| **Test Description: multi-site NS scale out triggered by a VNF/EM Request** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_MULTISITE\_004 | | | |
| **Test Purpose** | | To verify that a multi-site NS can be successfully scaled in (by removing VNF instances) if triggered automatically in MANO by a VNF/EM request | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.2.3 and 6.3.3)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to terminate virtualised resources * All VIM\_NFVIs support terminating virtualised resources * VNF/EM can send scale in request to MANO * MANO supports triggering multi-site scale in when the scale in request is received from VNF/EM * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * Multi-site NS is instantiated * Multi-Site NS has been scaled out by adding VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger the VNF/EM to send a scale in (by removing VNFs) request to MANO |  |
| 4 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated and not running in the correspondant VIM site / instance |  |
| 5 | | IOP Check | Verify that the impacted VNF related resources have been released by the proper VIM site / VIM instance |  |
| 6 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 7 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 8 | | IOP Check | Verify that the remaining VNF instance(s) have still multi-site connectivity through the multi-site VL(s) |  |
| 9 | | IOP Check | Verify that multi-site NS has been scaled in by running the end-to-end functional test |  |
| 10 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.10 VNF Scale Out in a multisite NS triggered by a MANO operator

| **Test Description: VNF scale out in a multi-site NS scale out triggered by a MANO operator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_VNF\_MULTISITE\_001 | | | |
| **Test Purpose** | | To verify that a VNF in a multi-site NS can be successfully scaled out (by adding VNFC instances(VMs)) when triggered by a MANO operator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE\_VNF  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE\_VNF | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA010 v2.3.1 (clauses 6.2.3 and 6.2.4)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request all VIM\_NFVIs to allocate virtualised resources * All VIM\_NFVIs supports allocating virtualised resources * MANO supports triggering scale out with an operator's action * MANO supports scale out by adding VNFC instances (VMs) * NS/VNF supports scale out by adding VNF instances (VMs) | | | |
|  | | | | | |
| Pre-test conditions | | * Multi-Site NS is instantiated * MANO is configured to trigger SCALE OUT (by adding VNFC instances) when triggered by a MANO operator | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger multi-site NS scale out (by adding VNFC instances (VMs) to a VNF in the NS) in MANO with an operator action |  |
| 3 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the additional VNFC instance(s) have been deployed according to the descriptors (VMs, VLs, CPs...) |  |
| 5 | | IOP Check | Verify that the additional VNFC instance(s) have been deployed according to the multi-site location constraints |  |
| 6 | | IOP Check | Verify that the additional VNFC instances(s) are running and reachable from the management network |  |
| 7 | | IOP Check | Verify that the additional VNFC instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 8 | | IOP Check | Verify that the additional VNFC instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 9 | | IOP Check | Verify that the addtional VNFC instance(s) have multi-site connectivity through the multi-site VL(s) |  |
| 10 | | IOP Check | Verify that NS has been scaled in by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.6.11 VNF Scale In in a multisite NS triggered by a MANO operator

| **Test Description: VNF scale in in a multi-site NS scale out triggered by a MANO operator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_VNF\_MULTISITE\_001 | | | |
| **Test Purpose** | | To verify that a VNF in a multi-site NS can be successfully scaled in (by removing VNFC instances(VMs)) when triggered by a MANO operator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_MULTISITE\_ENDPOINT\_SCALE\_VNF  SUT\_1\_NS\_1\_MULTISITE\_MIDDLEPOINT\_SCALE\_VNF | | | |
| **References** | | ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1 and 7.4.1)  ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA010 v2.3.1 (clauses 6.2.3 and 6.2.4)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.4) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to allocate virtualised resources * VIM\_NFVI supports terminating virtualised resources * MANO supports triggering scale in with an operator's action * MANO supports scale in by removing VNFC instances (VMs) * NS/VNF supports scale in by removing VNF instances (VMs) | | | |
|  | | | | | |
| Pre-test conditions | | * Multi-site NS is instantiated * Multi-site NS has been scaled out by adding VNFC instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger multi-site NS instantiation in MANO |  |
| 2 | | Preparation | Trigger multi-site NS scale out in MANO |  |
| 3 | | Stimulus | Trigger NS scale in (by removing VNFC instances (VMs) from a VNF in the NS) in MANO with an operator action |  |
| 4 | | Stimulus | Verify that the impacted VNFC instance(s) have been terminated and not running in the correspondant VIM site / instance |  |
| 5 | | IOP Check | Verify that the impacted VNFC instances related resources have been released by the proper VIM site / VIM instance |  |
| 6 | | IOP Check | Verify that the remaining VNFC instances(s) are still running and reachable through the management network |  |
| 7 | | IOP Check | Verify that the remaining VNFC instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 8 | | IOP Check | Verify that the remaining VNFC instance(s) have still multi-site connectivity through the multi-site VL(s) |  |
| 9 | | IOP Check | Verify that multi-site NS has been scaled in by running the end-to-end functional test |  |
| 10 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

## 3.7 Enhanced Platform Awareness

### 3.7.1 NS EPA Instantiation

| **Test Description: NS EPA instantiation** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_INSTANTIATION\_EPA\_001 | | | |
| **Test Purpose** | | To verify that an NS can be successfully instantiated with EPA requirements | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT  SUT\_1\_NS\_1\_MIDDLEPOINT | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.3)  ETSI GS NFV-IFA 005 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.2.3)  ETSI GS NFV-IFA 008 v2.3.1 (clause 6.2.3)  ETSI GS NFV-IFA 010 v2.3.1 (clause 6.3.2)  ETSI GS NFV-IFA 022 v0.8.0 (clause 5.2) | | | |
| **Applicability** | | * MANO can request VIM\_NFVIs to add a SW image * VIM\_NFVI support adding a SW image * MANO can request VIM\_NFVI to allocate virtualised resources * MANO supports deploying VNFs with EPA requirements * VIM\_NFVI supports allocating virtualised resources * VIM\_NFVI supports EPA attributes | | | |
|  | | | | | |
| **Pre-test conditions** | | * NSD, VLD(s), VNFFGD(s) and VNF Package(s) have been on-boarded in MANO * On-boarded VNFD(s) include EPA requirements * The software image repository is reachable by the VIMs * The required resources are available on the NFVIs | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Stimulus | Trigger NS instantiation in MANO |  |
| 2 | | IOP Check | Verify that the software images have been onboarded in the VIM |  |
| 3 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 4 | | IOP Check | Verify that the VNF (s) have been deployed according to the desciptors (VMs, VLs, CPs...) |  |
| 5 | | IOP Check | Verify that the VL and VNFFG instance(s) have been created according to the descriptors |  |
| 6 | | IOP Check | Verify that required EPA attributes have been configured as expected, e.g. checking configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 7 | | IOP Check | Verify that the VNF(s) are running and reachable through the management network |  |
| 8 | | IOP Check | Verify that the VNF(s) have been configured according to VNFD(s) (i.e by obtaining a result from the management interface) |  |
| 9 | | IOP Check | Verify that the VNF(s), VL(s) and VNFFG(s) have been connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that theNS is successfully instantiated by running the end-to-end functional test |  |
| 11 | | IOP Check | Verify that the EPA requirements are matched in the NS instance and the running VNFs (e.g. performance check) |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.2 NS with EPA requirements Scale Out triggered by MANO operator

| **Test Description: NS with EPA scale out triggered by MANO operator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_001 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically by a MANO operator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to allocate virtualised resources * VIM\_NFVI supports allocating virtualised resources * MANO supports triggering scale out with an operator's action * MANO supports deploying VNFs with EPA requirements * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger NS scale out (by adding VNF instances) in MANO with an operator action |  |
| 3 | | IOP Check | Verify that the scale out (by adding VNF(s)) procedure has been started in MANO |  |
| 4 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 5 | | IOP Check | Verify that required EPA attributes have been configured as expected, e.g. checking configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) have been deployed |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 9 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check) |  |
| 11 | | IOP Check | Verify that NS has been scaled out by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.3 NS with EPA requirements Scale In triggered by MANO operator

| **Test Description: NS with EPA scale out triggered by MANO operator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_001 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled in with EPA requirements (by removing VNF instances) if triggered automatically by a MANO operator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to terminate virtualised resources * VIM\_NFVI supports terminating virtualised resources * MANO supports triggering scale in with an operator's action * MANO supports managing VNFs with EPA requirements * MANO supports scale in by adding VNF instances * NS/VNF supports scale in by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated * NS has been scaled out by adding VNF instances with EPA requirements | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger NS scale in (by removing VNFs) in MANO with an operator action |  |
| 4 | | IOP Check | Verify that the scale in (by removing VNF(s)) procedure has been started in MANO |  |
| 5 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated |  |
| 6 | | IOP Check | Verify that EPA configurations of VNF(s) to be scaled-in have been deallocated/released as expected, e.g. checking de-configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 7 | | IOP Check | Verify that the impacted VNF related resources have been released by the VIM |  |
| 8 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 9 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 10 | | IOP Check | Verify that NS has been scaled in by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.4 NS with EPA requirements Scale Out triggered by a VNF Indicator

| **Test Description: NS with EPA scale out triggered by a VNF Indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_002 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically by a VNF Indicator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to allocate virtualised resources * VIM\_NFVI supports allocating virtualised resources * MANO supports receiving VNF indicators from VNF/EM * VNF/EM can send VNF indicator values to MANO * MANO supports triggering scale out when a given VNF Indicator value crosses a certain treshold * MANO supports deploying VNFs with EPA requirements * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger the VNF/EM to send the targeted VNF indicator to MANO until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the scale out (by adding VNF(s)) procedure has been started in MANO |  |
| 4 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 5 | | IOP Check | Verify that required EPA attributes have been configured as expected, e.g. checking configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) have been deployed |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 9 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check) |  |
| 11 | | IOP Check | Verify that NS has been scaled out by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.5 NS with EPA requirements Scale In triggered by a VNF Indicator

| **Test Description: NS with EPA scale in triggered by a VNF Indicator** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_002 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled in with EPA requirements (by removing VNF instances) if triggered automatically by a VNF Indicator | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to terminate virtualised resources * VIM\_NFVI supports termintaing virtualised resources * MANO supports receiving VNF indicators from VNF/EM * MANO supports managing VNFs with EPA requirements * VNF/EM can send VNF indicator values to MANO * MANO supports triggering scale in when a given VNF Indicator value crosses a certain treshold * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated * NS has been scaled out by adding VNF instances * MANO is configured to trigger SCALE IN (by removing VNF instances) when a given VNF Indicator value crosses a certain threshold | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger the VNF/EM to send the targeted VNF indicator to MANO until the configured threshold is crossed |  |
| 4 | | IOP Check | Verify that the scale in (by removing VNF(s)) procedure has been started in MANO |  |
| 5 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated |  |
| 6 | | IOP Check | Verify that EPA configurations of VNF(s) to be scaled-in have been deallocated/released as expected, e.g. checking de-configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 7 | | IOP Check | Verify that the impacted VNF related resources have been released by the VIM |  |
| 8 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 9 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 10 | | IOP Check | Verify that NS has been scaled in by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.6 NS with EPA requirements Scale Out triggered by a VIM KPI

| **Test Description: NS with EPA scale out triggered by a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_003 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to allocate virtualised resources * VIM\_NFVI supports allocating virtualised resources * MANO supports receiving VR related KPIs from VIM * VIM\_NFVI can send VR related KPIs to MANO * MANO supports triggering scale out when a given KPI crosses a certain treshold * MANO supports deploying VNFs with EPA requirements * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated * MANO is configured to trigger SCALE OUT (by adding VNF instances) when a given VIM KPI value crosses a certain threshold | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger the VIM to send the targeted KPI to MANO until the configured threshold is crossed |  |
| 3 | | IOP Check | Verify that the scale out (by adding VNF(s)) procedure has been started in MANO |  |
| 4 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 5 | | IOP Check | Verify that required EPA attributes have been configured as expected, e.g. checking configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) have been deployed |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 9 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check) |  |
| 11 | | IOP Check | Verify that NS has been scaled out by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.7 NS with EPA requirements Scale In triggered by a VIM KPI

| **Test Description: NS with EPA scale in triggered by a VIM KPI** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_003 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled in with EPA requirements (by removing VNF instances) if triggered automatically by a VIM KPI | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to terminate virtualised resources * VIM\_NFVI supports termintaing virtualised resources * MANO supports receiving VR related KPIs from VIM * MANO supports managing VNFs with EPA requirements * VIM\_NFVI can send VR related KPIs to MANO * MANO supports triggering scale in when a given KPI crosses a certain treshold * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated * NS has been scaled out by adding VNF instances * MANO is configured to trigger SCALE IN (by removing VNF instances) when a given VIM KPI value crosses a certain threshold | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger the VIM to send the targeted KPI to MANO until the configured threshold is crossed |  |
| 4 | | IOP Check | Verify that the scale in (by removing VNF(s)) procedure has been started in MANO |  |
| 5 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated |  |
| 6 | | IOP Check | Verify that EPA configurations of VNF(s) to be scaled-in have been deallocated/released as expected, e.g. checking de-configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 7 | | IOP Check | Verify that the impacted VNF related resources have been released by the VIM |  |
| 8 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 9 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 10 | | IOP Check | Verify that NS has been scaled in by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.8 NS with EPA requirements Scale Out triggered by a VNF/EM request

| **Test Description: NS with EPA scale out triggered by a VNF/EM request** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_OUT\_EPA\_004 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled out with EPA requirements (by adding VNF instances) if triggered automatically by a VNF/EM request | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to allocate virtualised resources * VIM\_NFVI supports allocating virtualised resources * VIM\_NFVI can send scale out request to MANO * MANO supports triggering scale out when the scale out request is received from VNF/EM * MANO supports deploying VNFs with EPA requirements * MANO supports scale out by adding VNF instances * NS/VNF supports scale out by adding VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Stimulus | Trigger the VNF/EM to send a scale out (by adding VNFs) request to MANO |  |
| 3 | | IOP Check | Verify that the scale out (by adding VNF(s)) procedure has been started in MANO |  |
| 4 | | IOP Check | Verify that the requested resources have been allocated by the VIM according to the descriptors |  |
| 5 | | IOP Check | Verify that required EPA attributes have been configured as expected, e.g. checking configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 6 | | IOP Check | Verify that the additional VNF instance(s) have been deployed |  |
| 7 | | IOP Check | Verify that the additional VNF instances(s) are running and reachable from the management network |  |
| 8 | | IOP Check | Verify that the additional VNF instances(s) have been configured according to the descriptors (i.e. by geting a result through the management interface) |  |
| 9 | | IOP Check | Verify that the additional VNF instances(s), VL(s) and VNFFG(s) are connected according to the Descriptors |  |
| 10 | | IOP Check | Verify that the EPA requirements are matched in the scaled VNF(s) (e.g. performance check) |  |
| 11 | | IOP Check | Verify that NS has been scaled out by running the end-to-end functional test |  |
| 12 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |

### 3.7.9 NS with EPA requirements Scale In triggered by a VNF/EM request

| **Test Description: NS with EPA scale in triggered by a VNF/EM request** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | | TD\_NFV\_NSLCM\_SCALE\_IN\_EPA\_004 | | | |
| **Test Purpose** | | To verify that a NS can be successfully scaled in with EPA requirements (by removing VNF instances) if triggered automatically by a VNF/EM request | | | |
| **Configuration** | | SUT\_1\_NS\_1\_ENDPOINT\_SCALE  SUT\_1\_NS\_1\_MIDDLEPOINT\_SCALE | | | |
| **References** | | ETSI GS NFV-IFA 013 v2.3.1 (clause 7.3.4)  ETSI GS NFV-IFA 005 v2.3.1 (clause 5.3.4)  ETSI GS NFV-IFA 006 v2.3.1 (clause 7.3.1, 7.4.1)  ETSI GS NFV-IFA 008 v2.3.1 (clause 7.2.4)  ETSI GS NFV-IFA 010 v0.8.0 (clause 6.2.3, 6.3.3) | | | |
| **Applicability** | | * MANO can request VIM\_NFVI to terminate virtualised resources * VIM\_NFVI supports terminating virtualised resources * VIM\_NFVI can send scale in request to MANO * MANO supports triggering scale in when the scale in request is received from VNF/EM * MANO supports managing VNFs with EPA requirements * MANO supports scale in by removing VNF instances * NS/VNF supports scale in by removing VNF instances | | | |
|  | | | | | |
| **Pre-test conditions** | | * NS is instantiated * NS has been scaled out by adding VNF instances | | | |
|  | | | | | |
| **Test Sequence** | **Step** | | **Type** | **Description** | **Result** |
| 1 | | Preparation | Trigger NS instantiation in MANO |  |
| 2 | | Preparation | Trigger NS scale out in MANO |  |
| 3 | | Stimulus | Trigger the VNF/EM to send a scale in (by removing VNFs) request to MANO |  |
| 4 | | IOP Check | Verify that the scale in (by removing VNF(s)) procedure has been started in MANO |  |
| 5 | | IOP Check | Verify that the impacted VNF instance(s) have been terminated |  |
| 6 | | IOP Check | Verify that EPA configurations of VNF(s) to be scaled-in have been deallocated/released as expected, e.g. checking de-configuration of:  - SR-IOV  - CPU pinning  - NUMA topology  - PCI passthrough  - .... |  |
| 7 | | IOP Check | Verify that the impacted VNF related resources have been released by the VIM |  |
| 8 | | IOP Check | Verify that the remaining VNF instances(s) are still running and reachable through the management network |  |
| 9 | | IOP Check | Verify that the remaining VNF instances(s), VL(s) and VNFFG(s) are still connected according to the descriptors |  |
| 10 | | IOP Check | Verify that NS has been scaled in by running the end-to-end functional test |  |
| 11 | | Termination | Trigger NS termination in MANO |  |
| **IOP Verdict** |  | | | |  |