

Oracle® Communications

Virtual Network Functions Manager

Installation and User Guide



Release 6.0.2.2.0
G42482-01
September 2025

ORACLE®

Copyright © 2019, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

1	Introduction	
1.1	References	1
1.2	Acronyms and definitions	1
1.3	Terminology	2
1.4	Limitations	2
1.5	My Oracle Support	3
2	Virtual Network Functions Manager Overview	
2.1	Advantage of Using VNFM	2
3	VNFM Lifecycle Management Interfaces	
4	VNFM OpenStack Prerequisites	
4.1	Prerequisite for Multiqueue on IPFE	3
4.2	Enabling the Multiqueue Feature on IPFEs	3
5	Install and Configure VNFM	
5.1	Access VNFM Using the REST Interface	7
5.2	VNFM Redundancy	8
5.3	Configurable Server Affinity Policy	8
5.4	VNFC Nomenclature	9
5.5	Supported VNFs by VNFM	9
6	Uninstalling VNFM	
7	Upgrading VNFM	
7.1	VNFM Upgrade Prerequisites	1
7.2	VNF Upgrade by VnfInstanceId	1

7.3	All VNF Upgrades	3
7.3.1	VNF State Diagram	4

8 VNFM User Management

8.1	Access Control in VNFM	1
8.2	Login to VNFM	2
8.3	Register to VNFM	3
8.4	Query All User Instances	4
8.5	Change Password of the User	5
8.6	Provision and Add User to VNFM	6

9 VNFM State Information

9.1	Change VNFM State	1
9.1.1	VNFM State Parameter Description	3
9.2	Making VNFM as a Standalone VNFM	3
9.3	Changing the Default Retry Configuration	4
9.4	API Versions	5

10 Deploying VNFs

10.1	Create a VNF Instance	2
10.2	Query VNF Instance	7
10.2.1	Query Individual VNF Instance	7
10.2.2	Query All VNF Instances (VNFM-V1)	9
10.2.3	Query all VNF Instances Using Filter	11
10.3	Modify VNF Information Operation	13
10.4	Deleting a VNF Instance	15
10.5	Grant Lifecycle Operation (VNFM - V2)	16
10.6	Instantiating VNFM Secondary	16
10.6.1	Sample Request for DYNAMIC IP Model	17
10.6.2	Sample Request for FIXED IP Model	18
10.6.2.1	Dynamic and Fixed IP Deployment Parameter Description	21
10.7	Instantiating the Network OAM VNF	22
10.8	Instantiating the DR Network OAM VNF (VNFM-V1)	30
10.8.1	Determining the DR NOAM XMI Resource IDs	31
10.9	Multiple XSI Support (1 - 16 XSI interfaces) - VNFM V1	37
10.10	Instantiating the Signaling VNF with Multiple XSI (1, 2, and 4 XSI Interface)	41
10.10.1	DSR Signaling VNF with Multiple XSI Support (1, 2 and 4 xsi interface only)	44
10.10.2	Instantiating the Signaling VNF with Service MP deployment	87
10.11	Instantiating Multiple Signaling VNFs	92

10.12	Instantiating the IDIH VNF	92
10.13	Instantiating the SDS Network OAM VNF	96
10.14	Instantiating the SDS DR Network OAM VNF	102
10.14.1	Determining the SDS DR NOAM XMI Resource IDs	103
10.15	Instantiating the SDS Signaling VNF	108
10.15.1	Determining the Signaling IMI Resource IDs	110
10.15.2	Determining the SDS NOAM XMI Resource IDs	110
10.16	Instantiating the ATS Master VNF (VNFM-V1)	119
10.17	Non-ConfigDrive VNF Instantiation	125
10.18	Scale VNF to Level (Only Scale Out)	126
10.18.1	Scale VNF to Level using InstantiationLevelId for DSR Signaling	127
10.18.2	Scale VNF to Level using ScaleInfo	132
10.18.3	IPFE and Service Mp Scale from Sizing 0	137
10.19	Operate VNF	138
10.19.1	Operate VNF Parameter Descriptions	141

11 VNF Instantiation across Multi Cloud / Multi Tenant / Multi Network

12 Heal VNF

12.1	Query Heal VNF Requests	2
12.1.1	Heal VNF Parameter Descriptions	3

13 Discover Stack

14 Query LCM Operation

14.1	Query Individual LCM Operation	1
14.2	Query All LCM Operation	7
14.3	Query All LCM Operation Using Filter	10

15 Terminating a VNF

15.1	Forceful Termination	1
15.2	Graceful Termination	2

16	Segregation of OAM and Replication Traffic Support (VNFM - V1)	
16.1	Sample request for DYNAMIC IP model of NOAM VNF in Dual Subnet	3
16.2	Sample request for DYNAMIC IP model of NOAM VNF in Single Subnet	5
16.3	Sample request for FIXED IP model of NOAM VNF in Dual Subnet	6
16.4	Sample request for FIXED IP model of NOAM VNF in Single Subnet	9
16.5	Sample request for Signaling flavor DIAMETER with multiple xsi for dynamic IP in Dual Subnet deployment model	11
16.6	Sample request for Signaling flavor DIAMETER with multiple xsi for dynamic IP in Single Subnet deployment model	14
16.7	Sample request for Signaling flavor DIAMETER with multiple xsi for fixed IP in Dual Subnet deployment model	16
16.8	Sample request for Signaling flavor DIAMETER with multiple xsi for fixed IP in Single Subnet deployment model	21
16.9	Sample request for DSR DR NOAM for Dynamic IP in Mix Mode Subnet deployment model	24
16.10	Sample request for DSR DR NOAM for Fixed IP in Dual Subnet deployment model	26
17	Changing the Default Configurations	
17.1	Changing Profile Name	2
17.2	Enable VNFM Logs with Different Log Levels (DEBUG, TRACE, WARN, ERROR)	2
17.3	Changing the Full Semantic Validation Configuration	2
18	Subscription Operation (VNFM - V2)	
19	VNFM SNMP ALERTS	
19.1	VNFM Alarms	3
19.2	VNFM MIB File	13
20	Import HTTPS/SSL Certificate into VNFM	
20.1	Recombine Existing PEM Keys and Certificates into VNFM	1
20.2	Copy Created Certificate (vnfm_default.jks) into VNFM	2
20.3	VNFM Self Signed Certificate Generation	2
21	Multiple HTTPS/SSL Certificate Support	
21.1	Configurable Keystore	2

22 NOAM IPv6 Migration

23 Troubleshooting VNFM

23.1	Debug VNFM	1
23.2	Adding Route for a New VIM	1
23.3	Reboot Tomcat	1
23.4	Resolve HA Alarms on VNF through VNFM Deployed Setup	2
23.5	Adding a Port in Openstack Security Groups	3
23.6	Debug SNMP System Alerts	3
23.7	Configure Flavor and Instantiation Levels in VNFM	3
23.8	Configuring DNS IP in VNFM	4
23.9	Disable SSH Weak Key Exchange Algorithms	5
23.10	Add or Update Certification	6
23.11	Changing VNF CLI and DSR GUI Log in Credentials	7

What's New in This Guide

This section introduces the documentation updates for release 6.0.2.2.0.

Release 6.0.2.2.0 - G42482-01, September 2025

- The following updates are made in the [Discover Stack](#) section:
 - Updated [Discover Stack](#) to specify that the VNFM supports backward compatibility from 5.4.x to 6.x using the Discover API operation.
 - Updated the version details in the Sample Request for Discover Interface.
- Updated the [Graceful Termination](#) section to specify that the VNFM supports automatic cleanup of signalling and DR NOAM data from NOAM.

1

Introduction

This document defines and describes the **DSR Virtual Network Functions Manager (DSR VNFM)**. DSR VNFM is an application that helps in the quick deployment of virtual DSRs by automating the entire deployment process and making it ready to use in the shortest possible time.

The VNFM is responsible for the lifecycle management of virtual network functions (VNFs) under the control of the network function virtualization orchestrator (NFVO).

1.1 References

- DSR Cloud Benchmarking Guide
- Or-VNFM Interface defined by ETSI NFV-SOL 003
- Import a Swagger Specification/Swagger UI
- DSR Cloud Install Guide
- DSR IP Flow Document
- DSR IPv6 Migration Guide

1.2 Acronyms and definitions

An alphabetized list of acronyms used in the document.

Table 1-1 Acronyms and definitions

Acronym	Definition
DA-MP	Diameter Agent Message Processor
DB	Database
DR	Disaster Recovery
DSR	Diameter Signaling Router
ETSI	European Telecommunications Standards Institute
GUI	Graphical User Interface
HA	High Availability
IDIH	Integrated Diameter Intelligence Hub
IP	Internet Protocol
IPFE	IP Front End
LCM	Lifecycle Management
MANO	Management and Orchestration
MP	Message Processing or Message Processor
NFVO	Network Functions Virtualization Orchestrator
NOAM	Network Operations and Maintenance
OAM	Operations, Administration, and Maintenance
OHC	Oracle Help Center

Table 1-1 (Cont.) Acronyms and definitions

Acronym	Definition
OSDC	Oracle Software Delivery Cloud
REST	Representational State Transfer
SOAM	System Operations and Maintenance
STP-MP	Signaling Transfer Point Message Processor
TSA	Target set Address
UDR	Usage Detail Records
UI	User Interface
VDSR	Virtual Diameter Signaling Router
VM	Virtual Manager
VNFM	Virtual Network Functions Manager
VNF	Virtual Network Functions
XMI	External Management Interface
XSI	External Signaling Interface

1.3 Terminology

This section describes terminologies used within this document.

Table 1-2 Terminologies and Definitions

Term	Definition
OpenStack controller	OpenStack controller controls the selected OpenStack instance.
Postman	A tool for creating REST requests.
Swagger UI	Swagger UI allows the users to interact with the API resources.
VNF instances	VNF instances are represented by the resources. Using this resource, the client can create individual VNF instance resources, and to query VNF instances.

1.4 Limitations

- Data present in Primary VNFM other than in external mounted volume are not synchronized with Secondary VNFM. For any manual process restart or configuration, data changes made on Primary VNFM must be done on Secondary VNFM.
- Scale-In feature is not supported.
- Terminate VNF deletes the entire stack and is not applicable for terminating a single server.
- Discover VNF stack supports:
 - Stacks that are created by using VNFM templates.
 - Stacks that are created by using same VNFM release.

- The stack created by VNFM GUI, Double Failure of Active VNFM and its Persistent volume.
- Inter version discovery is not supported. Stack can go into inconsistent state.
- Diameter Configuration is required for running the traffic.
- VNFM upgrade is not supported.

1.5 My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:
 - For Technical issues such as creating a new Service Request (SR), select **1**.
 - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

2

Virtual Network Functions Manager Overview

VNFM automates lifecycle operations for VNFs, as each VNF is managed independently.

To deploy DSR it requires creating and instantiating minimum two VNFs:

- Network OAM VNF
- Signaling VNF

Signaling VNFs can be instantiated any time after the network OAM has been instantiated.

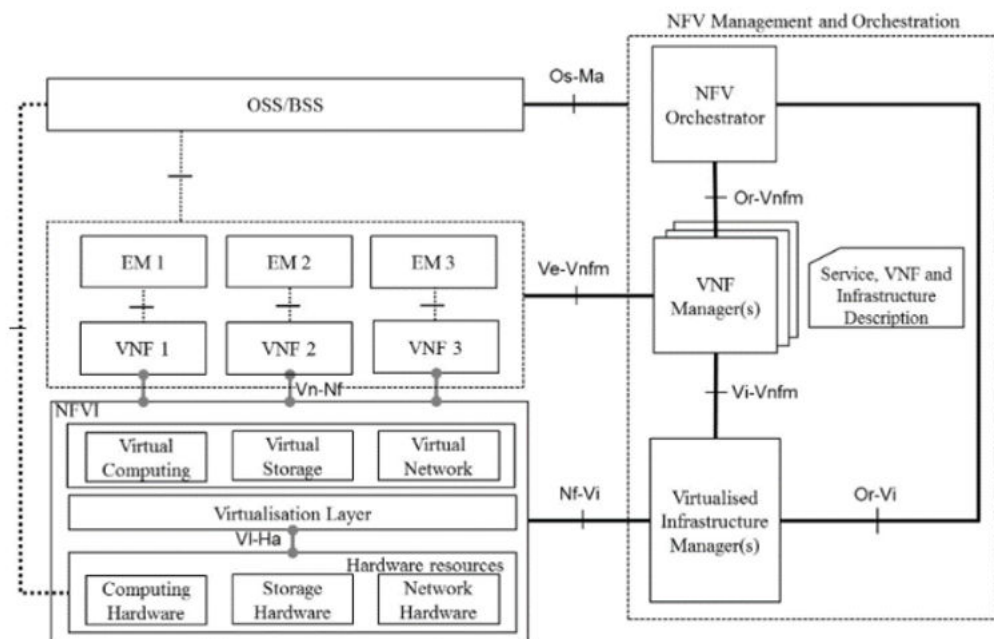
The main objective of the DSR VNFM is to provide an ETSI-compliant VNFM manager. VNFM would be beneficial in the following ways:

- Automating lifecycle management (LCM) operations for DSR VNFs. Automation of these operations can reduce their execution time.
- Providing a standardized interface to easily integrate with automation clients, especially ETSI-compliant NFVOs. The DSR VNFM provides a REST API that complies with ETSI NFV-SOL 003.

The VNFM is also helpful in responding quickly to changing customer requirements and delivers solutions for those requirements in a very short time.

The following figure illustrates the interaction between various components of DSR and VNFM:

Figure 2-1 ETSI MANO Specification



2.1 Advantage of Using VNFM

Deployment of Virtual DSR (vDSR) is performed by the following methods that requires manual processing:

- VM creation and installation process
- HEAT Template based installation (HEAT templates require manual updates)

The manual deployment consumes multiple hours to deploy a fully operational DSR and the HEAT template based installation needed more caution since it requires more manual work.

Using DSR VNFM, users can deploy an operational DSR on OpenStack within 20 minutes.

This application benefits both the internal and external customers by reducing operating expenses associated with the implementation and by reducing human errors by eliminating manual intervention.

VNFM V1:

The Current VNFM supports both VNFM v1 and v2, VNFM v1 is the existing VNFM.

Swagger URL: <https://<<VNFM HOST IP>>:8443/docs/vnfm/v1/>

VNFM V2:

VNFM comes with new version of VNFM i.e; VNFM - V2 which supports Cloud Service Archive (CSAR) based VNF deployment. It is compliant with ETSI spec 003 3.6.1. This is Orchestrator mode of deployment for both VNFM and VNFs.

Note

- To Instantiate VNF operation, required details need to be provided from Orchestrator and the sample "**vnfConfigurableProperties Json**" are given below for all the Instantiate VNF scenarios of VNFM - V2.
- For other APIs like Scale, Heal, Operate and Modify, the Additional Parameter Json for VNFM V2 is same as VNFM V1 and these details need to be provide through Orchestrator as VNFM V2 is Orchestrator mode of deployment.

Swagger URL: <https://<<VNFM HOST IP>>:8443/docs/vnfm/v2/>

vnfm_version: Based on API request on particular operation for VNFM v1 or v2, the vnfm_version will change in URL.

Example: URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<VNF ID received from create request>/instantiate

VNFM v1: URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/<VNF ID received from create request>/instantiate

VNFM v2: URL: https://<<VNFM HOST IP>>:8443/vnflcm/v2/vnf_instances/<VNF ID received from create request>/instantiate

3

VNFM Lifecycle Management Interfaces

The VNFM Lifecycle Management (LCM) interface supports the following operations:

- Create VNF
- Delete VNF
- Modify VNF
- Instantiate VNF
- Query Individual / All VNF(s)
- Scale VNF
 - Scale VNF to Level (Scale Out C Level servers of Signaling VNF)
 - Scale VNF to Arbitrary size (Scale Out C Level servers of Signaling VNF)
- Query Individual / All LCM Operation(s)
- Operate VNF
- Heal VNF
 - Query Heal VNF Requests
- Terminating VNF
- Discover VNF - Not part of ETSI standard
- Change VNFM State
- Query VNFM State
- Subscription API

4

VNFM OpenStack Prerequisites

Following are the prerequisites for using the VNFM:

- An OpenStack instance, TRAIN version. This OpenStack must be stable for cloud-init completion and proper network connectivity.
- Fixed IP's used for stack creation must be reachable.
- Security ports must be opened from the cloud infrastructure even if the VNFM template opens the port while deployment. However, it should not be blocked from cloud.
- Network used for stack deployment must be proper and "GATEWAY IP" should be present for each network.
- One OpenStack tenant per DSR signaling VNF. The DSR network OAM VNF may share a tenant with one of the signaling VNFs, if allowed.

Note

The openstack instance must have admin privileges for multi-tenant deployments.

- A DSR VM image must be in VMDK format as per GA release, named as: DSR-9.0.2.0.0-*.vmdk
Use sections to add and organize related content if another section heading is needed.
Where DSR-9.0.2.0.0-*.ova is the name of the OVA image delivered with the DSR build.
This image must be accessible from every tenant where VMs are deployed.
- VNFM assumes that the following flavors are defined on each OpenStack tenant on which the VMs are deployed.

For information about VNFM installation on OpenStack, see [Install and Configure the DSR VNFM](#).

Table 4-1 DSR Specific Flavors and Respective VNFM Types

VNFC Type	Image Name	Flavor Name	Minimum vCPUs	Minimum RAM (GB)	Minimum Disk (GB)
NOAM, DSR-DBSERVER, DSR-DR-NOAM	DSR-9.0.2.0.0.vmdk	dsr.noam (Regular Profile)	4	14	120
		dsr.noam (Large Profile)	8	14	120
SOAM		dsr.soam (Regular Profile)	4	14	120
		dsr.soam (Large Profile)	8	14	120
DA-MP		dsr.da dsr.da(Large Profile)	12	16	120
IPFE		dsr.ipfe	18	24	120

Table 4-1 (Cont.) DSR Specific Flavors and Respective VNFM Types

VNFC Type	Image Name	Flavor Name	Minimum vCPUs	Minimum RAM (GB)	Minimum Disk (GB)
STP-MP		dsr.vstp Note: Only if using vSTP function	6	16	120
SERVICE-MP		dsr.ServiceMp Note: Only if using vSTP function	8	12	120
SBR		dsr.sbr	12	25	120
SDS-NOAM, SDS-QS, SDS-DR-NOAM, SDS-DR-QS	SDS-9.0.2.0.0.vmdk	sds.noam	4	32	350
SDS-SOAM		sds.dpsoam	4	12	175
SDS-DP		sds.dp	6	10	175
DSR-IDIHAPP	apps-9.0.2.0.0.vmdk	appl-idih	4	8	64
DSR-IDIHMEDIATION	mediation-9.0.2.0.0.vmdk	med-idih	4	8	70
DSR-IDIHDB	oracle-9.0.2.0.0.vmdk	db-idih	4	8	220
UDR	UDR-14.0.2.0.0.vmdk	EIR	18	70	400
		VMNP	32	120	800
ATS Master	ats-9.0.2.0.0.	ats.master	4	16	1024

For more information about flavor, see section *DSR VM Configurations* of *DSR Benchmarking guide* for the minimum resource requirement with respect to each VNF flavor.

Note

To deploy a larger profile, the VM user needs to create the respective flavor in OpenStack.

Table 4-2 OpenStack VIM Connection Information

Parameter	Definition	Example
vimId	The identifier of the VIM instance. This identifier is managed by the NFVO.	"vimid"
vimType	The value of this attribute determines the structure of the "interfaceInfo" and "accessInfo" attributes, based on the type of the VIM.	"ETSINFV.OPENSTACK_KEYSTONE.V_3"
endpoint	VIM controller Identity API URI . The url representing the interface endpoint.	"https://mvl-dev1.us.oracle.com:5000/v3"
username	The username to use for access.	"*****"
password	Password to verified credentials for openstack controller	"*****"

Table 4-2 (Cont.) OpenStack VIM Connection Information

Parameter	Definition	Example
userDomain	The OpenStack user domain to use for the VIM connection.	"default"
projectDomain	The OpenStack project domain to use for the VIM connection.	"*****"
project	The OpenStack project to use for the VIM connection.	"*****"
region	The OpenStack region to use for the VIM connection.	"RegionOne"

VNFM adds a list of generic ports as a part of Openstack Security Groups. If traffic needs to be allowed through any other specific port, then that port must be added in Openstack Security Groups. For details about Adding a port in Openstack Security Groups, see [Adding a Port in Openstack Security Groups](#).

4.1 Prerequisite for Multiqueue on IPFE

To enable the *Multiqueue* feature on IPFE VMs, update the IPFE image and flavor. Perform the following procedure before creating IPFEs.

1. Run the following command to create a duplicate DSR image for IPFE with a different name on OpenStack:

```
openstack image create --file <Image File> --disk-format vmdk <Image Name>
```

Example:

```
openstack image create --file DSR-9.0.2.0.0.vmdk --disk-format vmdk
DSRIPFE-8.5.1.0.0.vmdk
```

2. Run the following command to update the multiqueue feature in the IPFE image:

```
glance image-update <IMAGE_ID> --property hw_vif_multiqueue_enabled=true
```

3. Run the following command to update the multiqueue feature in IPFE flavor:

```
openstack flavor set <IPFE_FLAVOR> --property
hw:vif_multiqueue_enabled=true
```

4.2 Enabling the Multiqueue Feature on IPFEs

After the deployment of DSR on IPFEs is complete, you can perform the following procedure to enable the *Multiqueue* feature to increase the performance of IPFE.

Note

- This procedure is not applicable for DSR 8.5 and later.
- This procedure is applicable for DSR 8.5 or earlier if the prerequisites, that is, updating the IPFE image and flavor, for the *Multiqueue* feature is complete.

- Ensure that the DSR deployment on IPFEs is complete.
 - Update the IPFE image and flavor as described in [Prerequisite for Multiqueue on IPFE](#).
1. Add the following lines to the network script of the interface that you want to change:

```
DEVICE=eth
TYPE=Ethernet
ETHTOOL_OPTS="-L ${DEVICE} combined <no_of_vCPUs>
```

For example, to set the number of queues to number of vCPUs, edit `/etc/sysconfig/network-scripts/ifcfg-eth_interface` to set the multiqueue value to the number of vCPUs.

2. Run the `service network restart` command as root user to restart the network.
3. Run the `ethtool -l <eth_interface>` command to check the setting.

5

Install and Configure VNFM

Perform the following procedure to install and configure VNFM:

Prerequisite

- Download and install pip.
- 1. Get the linux box which has installed OpenStack client. If not, install Python and PIP, then install the OpenStack client in linux box to interact with OpenStack through CLI.
Steps to install the OpenStack client.
 - a. Log in as a root user and run the `yum install python-devel` command.
 - b. Install OpenStack client by running the `pip install python-OpenStackclient` command.
 - c. The above command skips importing heatclient plugin, install the plugin by running the following command: `pip install python-heatclient`
- 2. Identify an OpenStack instance.

Note

The identified OpenStack instance must meet the [VNFM OpenStack Prerequisites](#).

- a. Download the OpenStack api credential file from OpenStack.
- b. Download the OpenStack RC file.
 - i. Log in to OpenStack GUI.
 - ii. Go to API Access section tab.
 - iii. Click download OpenStack RC file and download (Identity API v3) file.
- c. Source the downloaded OpenStack API RC file in linux box where OpenStack client is installed by running the following command: `source openrc.sh`
When prompted for the password, provide OpenStack controller password.
`/"]=]8fdz`
- 3. Download the HEAT templates for VNFM installation.

Note

Download the DSR VNFM (Atomic release example 4.0) HEAT templates and parameter files to run through the OpenStack CLI Client for internal use. The HEAT templates can be downloaded from the Dev Clear Case Server.
Example: dsratsa@nclde01

```
"/export/home/eagle/releases/TPD/prod/DSRVNFM/6.0.2.2.0/qcow2"
```

Go to the release folder, for example: VNFM_6.0.2.2.0_60.2.11.

The release folder contains the following files:

- VNFM_6.0.2.2.0_60.2.11 .qcow2 (VNFM image)
- vnfmArtifacts_VNFM_6.0.2.2.0_60.2.11.tar (tar file). Extract the tar file and find the VNFM template and the VNFM deployment parameter file.
- dsrVnfmVm.yaml (VNFM Template File)
- dsrVnfmParams.yaml (VNFM Parameter File)

4. Upload the image file to OpenStack:
 - a. From the OpenStack GUI, navigate to **Projects** and **Compute-Image**.
 - b. Click **Create Image**.
 - c. In the Create Image dialog box, select the suggested options for the following fields:
 - i. In the **Image Source** field, select **Image File**.
 - ii. In the **Image File** field, select the **VNFM 6.0.2.2.0 VM** image. The VNFM image can be obtained from Oracle Software Delivery Cloud (OSDC) Portal.
Image name:
VNFM_6.0.2.2.0_60.2.11.qcow2
 - iii. The **Minimum Disk** and **Minimum RAM** fields can be left blank.
 - d. The VNFM flavors must be provided with the appropriate values. For information about flavors, refer to the *DSR Cloud Benchmarking Guide*.
5. Create the VNFM Volume using:
 - a. The **OpenStack CLI**.
 - i. Create the VNFM volume to use as a part of the OpenStack. The VNFM supports a volume with the following specifications:
Volume size = 8 GB
Availability-zone = nova
For example: `OpenStack volume create --size 8 --availability-zone nova <Name of the volume>`
The above command displays the ID assigned to the newly created volume.
 - b. The **OpenStack GUI**:
 - i. Navigate to **Project > Volumes - Volumes**.
 - ii. Click **Create Volume**.
 - iii. In the Create Volume dialog box, perform as suggested for the following fields:

- iv. In the **Size (GiB)** field, give 8 as its size.
- v. In the **Availability Zone** field, give nova as its value.
- vi. Get the ID of the volume created above and update the `dsrVnfmVolumeId` parameter in the `dsrVnfmParams.yaml` file.

Note

- To change the images and flavors of VNFCs, configure the respective parameters in: `/opt/vnfm/config/VmInfo.xml`
- To change the default properties, configure the respective parameters in: `/opt/vnfm/config/VnfmProperties.xml`

- 6. Modify the input parameters:
 - a. Edit the HEAT template file `dsrVnfmParams.yaml`

Note

- The input parameters are given as key/value pairs. Modify only the values (the part to the right side of the colon).
- The formatting is an important factor in YAML file. Do not remove any leading spaces or add any lines to the file.
- While creating IPv4 setup of Vnfm (Vnfm network is IPv4), use DNS and NTP of IPv4 and while creating IPv6 setup of Vnfm (Vnfm Network is IPv6), use DNS and NTP of IPv6.

- b. Edit the values as per the guidelines provided in the following table:

Table 5-1 Parameters and Definitions for VNFM Installation

Parameter	Value
<code>dsrVnfmVmName</code>	Enter a name for the VM. Alphanumeric characters and "_" are allowed.
<code>dsrVnfmImage</code>	Enter the name of the image uploaded in the previous step.
<code>dsrVnfmFlavor</code>	Enter the name of a flavor that is loaded onto OpenStack.

Table 5-1 (Cont.) Parameters and Definitions for VNFM Installation

Parameter	Value
vnfmNetwork	Enter the name and subnet of a network that external clients can use to communicate with the VNFM. (The user can also give an IP along with the network in case of fixed IP deployment) (IPv6 or IPv4) Format for Dynamic IP deployment: <pre>"vnfmNetwork": { "network" : "<Network Name>", "subnet" : "<Subnet Name>" }</pre> Format for Fixed IP deployment: <pre>"vnfmNetwork": { "network" : "<Network Name>", "fixed_ip" : "<ip>" }</pre>
vimNetwork	Enter the name and subnet of a network that VNFM uses to route VIM traffic. (IPv4 or IPv6) Format for Dynamic IP deployment: <pre>"vimNetwork": { "network" : "<Network Name>", "subnet" : "<Subnet Name>" }</pre> Format for Fixed IP deployment: <pre>"vimNetwork": { "network" : "<Network Name>", "fixed_ip" : "<ip>" }</pre>
ntpServer	Enter the IP address of an NTP server with which the VNFM synchronizes the time. The OpenStack controller hosts an NTP server so the IP address of the OpenStack controller is usually a good value. Note: VNFM can support any of these NTP resources: IPv4, IPv6, or both IPv4 and IPv6 NTP entries.
dnsServer	Enter the IP address of DNS server with which VNFM resolves the hostname. For an IPv4 setup, enter IPv4 DNS server IP and for an IPv6, use IPv6 DNS server IP.
dsrVnfmAZ	Enter the availability zone to place the VNFM. The "nova" is the default availability zone and is usually the right value.
dsrVnfmVolumeld	Enter the volume name to use as persistence storage for the VNFM.
vimRouteAddress	Enter the OpenStack network address/subnet mask. This is going to be use communication between VNFM and OpenStack (Vim) network. User can provide the list of route address separated by comma.
snmpReceiverAddressPort (Optional)	IP and Port of the SNMP Trap Receiver/ SNMP Manager. Default: 127.0.0.1/162,::1/162 (Not required for IPv6 brackets.)

Note

- In case of fixed IP deployment for VNFM, the network name and IP must be given in the following syntax for `vnfmNetwork` or `vimNetwork` parameter in `dsrVnfmParams.yaml` file: `vnfmNetwork: {"network":"ext-net", "fixed_ip":"10.196.52.175"} vimNetwork: {"network":"ext-net2", "fixed_ip":"10.196.52.176"}`
- In case of dynamic IP deployment for VNFM, the network name and subnet should be given in the following syntax for `vnfmNetwork` or `vimNetwork` parameter in `dsrVnfmParams.yaml` file: `vnfmNetwork: {"network":"ext-net", "subnet": "ext-net-subnet"} vimNetwork: {"network":"ext-net2", "subnet": "ext-net2-subnet"}`
- User needs to give mandatory OpenStack network address `vimRouteAddress` parameter in `vnfm` parameter.
Syntax: `vimRouteAddress: <OpenStack Network address>/<subnet mask>`

For example

`vimRouteAddress: 10.75.167.0/24`

In case of list of OpenStack cloud:

`vimRouteAddress: 10.75.167.0/24, 10.75.185.0/24`

- If user needs to communicate with multiple OpenStack cloud using one `vnfm` then the user must provide multiple OpenStack network address while installing VNFM.

User can also add other OpenStack cloud network after installing `vnfm`, by performing the steps provided in section [Adding Route for a New VIM](#).

User must provide optional SNMP Manager IP and Port as `snmpReceiverAddressPort` parameter in `dsrVnfmParams.yaml` file.

Syntax: `snmpReceiverAddressPort: IP/PORT, IP/PORT`

For example:

In case of Dual SNMP Manager: `snmpReceiverAddressPort: 10.75.189.151/8900, 2606:b400:605:b813::5/7400`

In case of Single SNMP Manager: `snmpReceiverAddressPort: 2606:b400:605:b813::5/7400`

`ntpServer`: Users must provide mandatory `ntpServer` details which can support up to 3 `ntp` resources at a time in the `dsrVnfmParams.yaml` file. Example1:

`ntpServer: 10.250.32.10`, example 2: `ntpServer:`

`2606:b400:605:b912:200:5eff:fe00:1f7`, Ex3: `ntpServer:`

`10.250.32.10, 2606:b400:605:b912:200:5eff:fe00:1f7, 10.250.32.56`

c. Once editing is done, save the file.

7. Deploy the VNFM using the OpenStack CLI by running the following command:

```
OpenStack stack create -t dsrVnfmVm.yaml -e dsrVnfmParams.yaml <stackName>
```

8. To query the VNFM release details after VNFM deployment, run the following command:

```
$>[dsrvnm@releasevnfmvm /]$ sudo ./install_vnfm.py --info
[sudo] password for dsrvnm
```

VNFM release information:

```
Product Name : VNFM
Product Release : 6.0.2.2.0
DSR Supported Release(s) :
9.0.2.0.0
9.0.2.1.0
9.0.2.2.0
```

Refer to the following table while choosing the IP versions:

Note

The naming convention for ens3, ens4, enp3s0, or enp4s0 may have different names based on the interface configuration.

Table 5-2 IP Version Mapping

VNFM External IP Version (REST interface) ens3	VNFM Vim IP Version (VIM interface) ens4	OpenStack Controller VIM IP	DSR IP/ Other Traffic	Route ens3	Route ens4	Dual Snmp Manager Support	Comments
IPv6	IPv4	IPv4	IPv6 / IPv4	Default	Default	Supported	- Supported. - All the openstack traffic/packet goes through VIM IP (ens4). - IPv6 traffic other than VIM traffic goes through VNFM IP (ens3). - IPv4 traffic other than VIM traffic goes through VIM IP (ens4).
IPv4	IPv4	IPv4	IPv4	Default	Limited to VIM Network Routes	Supported	- Supported. - Only IPv4 traffic is supported. - MMI calls to IPv6 VNFs fails. VNFs are instantiated and Cloud init might be complete, but other LCM operations fail. - So, DSR, SNMP, NTP, DNS, and any other servers communicating with VNFM must be on IPv4 networks.
IPv4	IPv6	IPv6	IPv6 / IPv4	Default	Default	Supported	- All the openstack traffic/packet goes through VIM IP (ens4). - IPv4 traffic other than VIM traffic goes through VNFM IP (ens3). - IPv6 traffic other than VIM traffic goes through VIM IP (ens4).

Table 5-2 (Cont.) IP Version Mapping

VNFM External IP Version (REST interface) ens3	VNFM Vim IP Version (VIM interface) ens4	OpenStack Controler VIM IP	DSR IP/ Other Traffic	Routens3	Routens4	Dual Snmp Manager Support	Comments
IPv6	IPv6	IPv6	IPv6	Default	Limited to VIM Network Routes	Supported	- Only IPv6 traffic is supported. - MMI calls to IPv4 VNFs fails. VNFs are instantiated and Cloud init is complete, but other LCM operations fail. - So, DSR, SNMP, NTP, DNS, and any other servers communicating with VNFM must be on IPv6 networks.
IPv6	IPv4	IPv6					Not Applicable. The VIM IP version and the controller IP version are different. The communication never gets established.
IPv4	IPv4						
IPv4	IPv6	IPv4					
IPv6	IPv6						

Note

- **VNFM External IP Version (REST interface) ens3**- VNFM external IP interface to support the VNFM REST API.
- **VNFM Vim IP Version (VIM interface) ens4**- VNFM IP is used to communicate with the VIM controller. VNFM ens4 IP and VIM controller IP should be in the same IP version. Both of them should be in either IPv4 or IPv6.
- **OpenStack Controller VIM IP**- OpenStack controller VIM IP that creates the VNF through VNFM. Multiple OpenStack VIM controller IP can be provided during VNFM installation with VIM subnet.
- **DSR IP**- DSR IP is the VNF IP. VNFM eth0 IP communicates with DSR XMI interface for DSR Cloud init LCM operation. So, DSR xmi IP and VNFM eth0 IP must have the same IP version, either IPv4 or IPv6.

VNF's should be reachable from VNFM to support all VNFM operations.

Primary VNFM Installation VIA Orchestrator (VNFM - V2):

Primary vnf can be installed using NFVD via VIM mode of deployment.

5.1 Access VNFM Using the REST Interface

The VNFM is accessible using a REST interface. There is no provision to access the REST interface through CLI, or GUI, however it can be accessed through a Swagger specification provided for the REST interface. There are many other compatible interfaces that can be used with popular REST testing tools. Some of the most widely used tools that can be used with the REST testing tool are:

Swagger UI

With the **Swagger UI**, a GUI can be generated from the Swagger specification.

Swagger specifications can be found post VNFM installation at (https://<VNFM_IP>:8443/docs/vnfm/<<vnfm_version>>).

Postman

Another popular tool for creating REST requests is the **Postman** tool. It is available as a standalone app and as a Chrome browser plug-in. You can import a Swagger specification to allow Postman to understand the VNFM REST API in detail, which allows it to assist you while creating requests and interpreting responses.

5.2 VNFM Redundancy

VNFM supports the Redundancy model. It works with two servers in Primary - Secondary states. Data in the external mounted volume is synchronized from Primary to Secondary VNFM.

VNFM can have two states and a transitory third state as follows:

- Primary
- Secondary
- Transient

The Transient state occurs when a CHANGE VNFM STATE request fails due to network connectivity issues. In this case, the orchestrator must wait until the connectivity issue is resolved and re-triggers the CHANGE VNFM request.

Secondary VNFM can be created as any other regular VNF by using CREATE and INSTANTIATE VNF requests. It can be created only when the PRIMARY VNFM is a standalone VNFM.

State of VNFM can be queried using the QUERY VNFM STATE INFORMATION REST request.

VNFM state can be changed using the CHANGE VNFM STATE REST request.

Note

When VNFM is in Secondary or Transient state, it accepts all GET requests and only the following POST requests: Login To VNFM and Change VNFM State.

5.3 Configurable Server Affinity Policy

Server Affinity Policy configuration is supported during the VNF Instantiation of DSR/SDS VNF's only. This policy can be configured on VNFC level.

Scaling uses the same affinity policy provided during VNF Instantiation and hence affinity policy option is not required during scaling. Default Server Group affinity policy is "anti-affinity".

Allowed Policies

1. Anti Affinity: place instances on separate hosts.<Default>

2. Affinity: places instances on the same host.
3. Soft Anti Affinity: place instances on separate hosts if possible.
4. Soft Affinity: place instances on the same host if possible.

5.4 VNFC Nomenclature

The following table contains information about VNFC Nomenclature.

VNF Instance Name (max 22 Characters)	VNFC Type	Nomenclature (max 5 characters)	Server Name (VM Hostname) (max 30 Characters)
<User Input>	DSR NOAM	DNO	<user-input>-DNO00
<User Input>	DSR SOAM	DSO	<user-input>-DSO00
<User Input>	DSR DAMP	DMP	<user-input>-DMP00
<User Input>	DSR IPFE	DIP	<user-input>-DIP00
<User Input>	STP MP	STPMP	<user-input>-STPMP00
<User Input>	SBR (Session/Binding/Universal)	SBR	<user-input>-SBR00
<User Input>	UDR	UDR	<user-input>-UDR00
<User Input>	DSR DR NOAM	DDRNO	<user-input>-DDRNO00
<User Input>	SDS NOAM	SNO	<user-input>-SNO00
<User Input>	SDS QS	SQS	<user-input>-SQS00
<User Input>	SDS SOAM	SSO	<user-input>-SSO00
<User Input>	SDS DP	SDP	<user-input>-SDP00
<User Input>	SDS DR NOAM	SDRNO	<user-input>-SDRNO00
<User Input>	SDS DR QS	SDRQS	<user-input>-SDRQS00
<User Input>	DsrldihApp	IDAPP	<user-input>-IDAPP00
<User Input>	DsrldihMed	IDMED	<user-input>-IDMED00
<User Input>	DsrldihDb	IDDB	<user-input>-IDDB00
<User Input>	atsMaster	ATSMA	<user-input>-ATSMA00

5.5 Supported VNFs by VNFM

The table below contains a list of all the VNFs supported by VNFM:

Table 5-3 Supported VNFs and VMs

Supported Dynamic IP VNFs	Supported VNFCs	Support ed Dynam ic IP VNF	Support ed Fixed IP VNF	Support ed Dual Stack IP VNF	VNF Depend ency	Mixed Mode (XMI (Single/Dual), IMI(Sing le/Dual) and XSI-1, 2, 4(Single /Dual))	Mixed Mode XSI-1, 2, 4 (4XSI-1, 2, 3, 4) (Single/ Dual)
DSR NOAM	NOAM (Active/Standby)	Yes	Yes	Yes		Yes	N/A

Table 5-3 (Cont.) Supported VNFs and VMs

Supported Dynamic IP VNFs	Supported VNFCs	Support ed Dynamic IP VNF	Support ed Fixed IP VNF	Support ed Dual Stack IP VNF	VNF Depend ency	Mixed Mode (XMI (Single/Dual), IMI(Single/Dual) and XSI-1, 2, 4(Single/Dual))	Mixed Mode XSI-1, 2, 4 (4XSI-1, 2, 3, 4) (Single/Dual)
DSR DR NOAM	DR NOAM (Active/Standby)	Yes	Yes	Yes	DSR NOAM	Yes	N/A
DSR Signaling	SOAM (Active/Standby), DA-MP, STP-MP, IPFE, SBR, UDR	Yes	Yes	Yes (Only for DIAMETER, DIAMETER+SBR flavor)	DSR NOAM	Yes*	Yes*
IDIH	APP, MEDIATION, DB Server	Yes	Yes	No	DSR Signaling	No	N/A
SDS NOAM	NAOM (Active/Standby) and Query Server	Yes	Yes	Yes		Yes	N/A
SDS DR NOAM	DR NAOM (Active/Standby) and Query Server	Yes	Yes	Yes	SDS NOAM	Yes	N/A
SDS Signaling	SOAM (Active/Standby), DP Server	Yes	Yes	Yes	SDS NOAM	Yes	Yes
ATS Master	MASTER	Yes	Yes	No		No	No

Yes* -

- Single Subnet (either IPv4 or IPv6) - supported for all flavors.
- Dual subnet and Single Dual Mixed subnet mode - only DIAMETER, DIAMETER+SBR flavor supported.

The following table provides a sample combination of DSR-SOAM (for Diameter Flavor) of XSI network interfaces:

Table 5-4 Sample Combination of DSR-SOAM (for Diameter Flavor)

VNF TYPE	XSI-1	XSI-2	XSI-3	XSI-4
DSR Signaling (Diameter Flavor)	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6
	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6
	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6
	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6

The following table provides a sample combination of DSR-SOAM (for Diameter+SBR Flavor) of XSI, SBR network interfaces:

Table 5-5 Sample Combination of DSR-SOAM (for Diameter+SBR Flavor)

VNF TYPE	XSI1	XSI2	SBR
DSR Signaling (Diameter+SBR Flavor)	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6
	Single Stack IPv4 or IPv6	Dual Stack IPv4 and IPv6	Single Stack IPv4 or IPv6
	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6

The following table provides a sample combination of SDS-SOAM:

Table 5-6 Sample Combination of SDS-SOAM

VNF TYPE	XMI	IMI
SDS Signaling	Single Stack IPv4 or IPv6	Single Stack IPv4 or IPv6
	Single Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6
	Dual Stack IPv4 and IPv6	Single Stack IPv4 or IPv6
	Dual Stack IPv4 and IPv6	Dual Stack IPv4 and IPv6

6

Uninstalling VNFM

Note

VNFM uninstallation requires manual intervention and impacts existing configurations. Proceed only if you intend to completely remove the VNFM.

Following are the steps to uninstall VNFM:

1. Log in to your OpenStack cloud platform using appropriate credentials.
2. Delete the VNFM Stack: Within OpenStack, locate the management interface for managing deployed stacks. Identify the VNFM stack and initiate its deletion process.
3. Delete Associated Volume (Optional): After removing the VNFM stack, an associated volume containing VNFM data will remain. Deleting this volume permanently removes all VNFM data.

Note

The volume cannot be restored. If you want to save data for future VNFM deployments, don't delete the volume.

Preserving Configurations for Redeployment:

If you intend to reinstall VNFM in the future and require current data, save the associated volume during the uninstalling procedure. This volume can then be used during a new VNFM deployment to restore your previous settings.

Security Considerations for Redeployment:

The VNFM volume stores sensitive operational and configuration information. When reinstalling VNFM, the administrator is responsible for attaching this volume to the new instance. Ensure that data on the volume is only accessed through the VNFM application.

7

Upgrading VNFM

Perform the following procedure for both primary and secondary VNFM each one after the other:

1. Change the VNFM state to secondary.
2. The current VNFM stack must be deleted. All the data is stored in the volume which is created in the install procedure. This acts as a persistent storage, so the stack can be safely deleted - the volume will be automatically detached from the stack.
3. The user must follow the steps in the [Install and Configure VNFM](#) with the new IMAGE provided. Flavor, Volume need not be created again. The existing volume ID must be given as the volume ID in the `dsrVnfmParams.yaml` file.

Note

- VNFM supports both the fixed and dynamic IP support. In order to bring up the new VNFM with the same IP as the existing one, the user can use Fixed IP deployment model.
- If the existing volume that is going to be attached to other stack is full (around 7GB), then it will take some time to restart the VNFM and load the data.
- It supports with same versions of VNFM releases, major VNFM version upgrade is not supported. For Backward compatibility, user must follow [Discover Stack](#) feature.

7.1 VNFM Upgrade Prerequisites

Following are the prerequisites for the VNFM Upgrade:

- VNF upgrade is enabled from 5.4.x v1 to 6.0.2.1.0 v1.
- Since v2 based deployment is through NFVD, the v1 to v2 is not supported.

Post VNFM upgrade 5.4.x VNF files will be available in `/var/vnfm/instances/*`. The upgrade endpoint upgrades the VNF files and saves them in `/var/vnfm/instances/v1/*` path. After VNF upgrade is successful, old files from the following path `/var/vnfm/instances/` will be deleted.

7.2 VNF Upgrade by VnfmInstanceId

VNF Upgrade can be performed after all VNFs in the VNF are upgraded from 8.6.x version to 9.0.2, no LCMs are supported on partially upgraded VNF. Following endpoint can be invoked to upgrade individual VnfmInstance based on VnfmInstanceId from 5.4.x and 8.6.x to 6.0.2 and 9.0.2 version.

Upgrade Endpoint:

`https://<VNFM_HOST_IP>:8443/vnfm/v1/instances/<instanceId>/upgrade`

Note

Image parameters in the request body are optional. Latest Vnf Image versions will be picked up from `VmInfo.yaml` file if image parameters are missing in the request body.

VNF upgrade request

URL: `https://<VNFM_HOST_IP>:8443/vnflcm/v1/instances/<instanceId>/upgrade`

Accept: `application/json`

Content-Type: `application/json`

Example for DSR NOAM upgrade:

```
{
  "image": "DSR-9.0.2.0.0_99.13.0"
}
```

Example for DSR DR NOAM upgrade:

```
{
  "image": "DSR-9.0.2.0.0_99.13.0"
}
```

Example for DSR Signaling upgrade:

```
{
  "soamImage": "DSR-9.0.2.0.0_99.13.0",
  "daImage": "DSR-9.0.2.0.0_99.13.0",
  "ipfeImage": "DSR-9.0.2.0.0_99.13.0",
  "stpImage": "DSR-9.0.2.0.0_99.13.0",
  "serviceMpImage": "DSR-9.0.2.0.0_99.13.0",
  "sbrImage": "DSR-9.0.2.0.0_99.13.0"
}
```

Example for SDS Noam upgrade:

```
{
  "sdsNoamImage": "SDS-9.0.2.0.0_99.13.0",
  "sdsQsImage": "SDS-9.0.2.0.0_99.13.0"
}
```

Example for SDS DR Noam upgrade:

```
{
  "sdsDrNoamImage": "SDS-9.0.2.0.0_99.13.0",
  "sdsDrQsImage": "SDS-9.0.2.0.0_99.13.0"
}
```

Example for SDS Signaling upgrade:

```
{
  "dpSoamImage": "SDS-9.0.2.0.0_99.13.0",
  "dpImage" : "SDS-9.0.2.0.0_99.13.0"
}
```

Example for ATS Master upgrade:

```
{
  "atsMasterImage": "ats-9.0.2.0.0-1.0.28"
}
```

7.3 All VNF Upgrades

All VNF Upgrades:

The endpoint can be invoked to upgrade individual VnfInstance based on VnfInstanceId from 5.4.6 and 8.6.x to 6.0.2 and 9.0.2 version.

Upgrade Endpoint:

`https://<VNFM_HOST_IP>:8443/vnflcm/v1/instances/upgradeall`

Note

Image parameters are not enabled in upgradeall. Latest VNF image versions will be picked up from VmInfo.yaml file.

Upgrading VNF from VNFM

VNF upgrade is supported for the VNFs deployed using v1 version APIs. After VNFM is upgraded to 6.0.2 version.

Upgrade supported for the following VNFs:

- DSR NOAM
- DSR DR NOAM
- DSR Signaling
- SDS NOAM
- SDS DR NOAM
- SDS Signaling
- ATS Master

Upgrade not Supported for the following VNFs:

- APIGW
- IDIH
- ATS Core and Tools

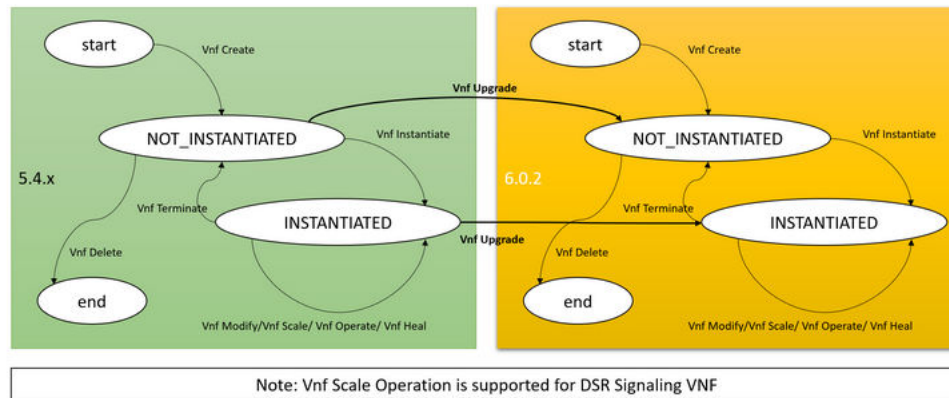
- PROV GW
- Secondary VNFM upgrade is not supported, it has to be deployed newly from primary VNFM.

7.3.1 VNF State Diagram

Following are the few high level scenarios based on state diagram:

- The VNF created on 5.4.x can be upgraded to 6.0.2 and all LCMs such as VnfInstantiate, VnfModify, VnfOperate, VnfHeal, VnfScale, VnfTerminate and VnfDelete can be performed
- The VNF created and Instantiated on 5.4.x can be upgraded to 6.0.2 and all LCMs such as VnfModify, VnfOperate, VnfHeal, VnfScale, VnfTerminate and VnfDelete can be performed.
- The VNF created and Instantiated and scaled on 5.4.x can be upgraded to 6.0.2 and all LCMs such as VnfModify, VnfOperate, VnfHeal, VnfScale, VnfTerminate and VnfDelete can be performed.

Figure 7-1 VNF State Diagram



8

VNFM User Management

(Required) Enter introductory text here, including the definition and purpose of the concept. The initial build is delivered with two pre-installed users that are admin and reader. The user must login to VNFM first using the given credentials to generate an **OAuth/X-Token** for the admin using the API "Login to VNFM".

The password of the admin must be changed using the generated **X-Token**, and a new password must be stored using the **Change Password** API.

The new users is registered using the **Register to VNFM** API.

Once the registration request is sent by the user, the admin has the access to view the registration request instance with the help of the **OAuth/X-Token** through the **Query all user instances** API.

The admin can provision the incoming requests and add the user request using the **Provision and Add** API.

Upon the successful registration, the user can simply login to VNFM using the credentials to generate an **OAuth/X-Token** and use it for other LCM-Operations.

8.1 Access Control in VNFM

The admin user has access to use the available APIs. However, the user with reader privileges, is restricted to use the following:

- Query All VNF Instances:

`https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnf_instances/`

- Query Individual VNF Instance:

`https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnf_instances/{vnfInstanceId}`

- Query All LCM Operation:

`https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs`

- Query Individual LCM Operation:

`https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/{vnfLcmOpOccId }`

- Query Heal VNF requests:

`https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vips/{vip}/heal`

- Query VNFM State Information:

`https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnfmState`

8.2 Login to VNFM

The user must provide the `grant_type` and the Authorization to generate an authentication token ergo OAuth-Token. Where,

- `grant_type`: `grant_type` of the registered user. Currently, supporting only `client_credentials`.
- Authorization: Authorization of the registered user. Currently, supporting Basic Format (Accepting Encrypted value of `"username':'password'"`).

Sample Request: Login to VNFM request generated

URL: `https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnfm_login/oauth/token`

Accept: `application/json`

Content-Type: `application/json`

Example for Login:

```
{
  "grant_type": "xxxx",
  "Authorization": "xxxx"
}
```

Sample Response: Login to VNFM Response

200 Created

Content-Type: `application/json`

Request URL: `https://<VNFM_HOST_IP>:8443/vnflcm/<<vnfm_version>>/vnfm_login/oauth/token`

```
"access_token":
"eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJRCBZiB0b2t1b2I6IHRva2VuX1Z0Rk0iLCJpYXQiOiJlODcyNTQ2MDAsInN1YiI6IiN1YmplY3Qgb2YgSldUIiwiaXNzIjoiaXNzdWVyaW9mIFRva2VuOiBPCmFjbGUiLCJ0eXkiOiJ0eXkiIiwiaWF0IjoiYXVkaWVudUg0IjBhZG1pb2I6ImV4cCI6MTY4NzI1ODIwMH0.2r8-JaaP2-YfbRRpmQ6Zyik3vzbKweofsuHsFnQoz-g",
  "token_type": "Bearer",
  "expires_in": "3600000",
  "refresh_token":
"eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJRCBZiB0b2t1b2I6IHRva2VuX1Z0Rk0iLCJpYXQiOiJlODcyNTQ2MDAsInN1YiI6IiN1YmplY3Qgb2YgSldUIiwiaXNzIjoiaXNzdWVyaW9mIFRva2VuOiBPCmFjbGUiLCJ0eXkiOiJ0eXkiIiwiaWF0IjoiYXVkaWVudUg0IjBhZG1pb2I6ImV4cCI6MTY4NzI1ODIwMH0.2r8-JaaP2-YfbRRpmQ6Zyik3vzbKweofsuHsFnQoz-g"
```

The following table describes the parameters used for sending request to VNFM:

Table 8-1 Parameters and Definitions

Parameter	Definitions
grant_type	client_credentials.
Authorization	base64 decode of "username:password".

Note

Authorization: Basic "YWRtaW46YWRtaW4=" where "YWRtaW46YWRtaW4=" is base64 encode of "username:password".

8.3 Register to VNFM

The new user must provide the username, the password and the access to send a successful registration request.

Note

A valid password must be in range between 8 to 31 characters, with at least one digit, at least one lowercase letter, at least one uppercase letter, at least one special character, and should not contain white spaces.

Sample Request: Register to VNFM request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnfm_register

Accept: application/json

Content-Type: application/json

Authorization: Token generated after login

Example for Registration:

```
{
  "username": "xxxx",
  "password": "xxxx",
  "access": "read/admin"
}
```

Sample Response: Register to VNFM Response

201 Created

Content-Type: application/json

Authorization: Token generated after login

Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnfm_register

```
{
```

"response": "Registration Request Sent"

The following table describes the parameters used for sending request to VNFM:

Table 8-2 Parameters and Definitions

Parameter	Definitions
username	Username of the new user.
password	password Password of the new user.
access	Scope of the new user.

8.4 Query All User Instances

The admin must provide the Authorization to view all the incoming registration requests.

Note

Only the admin has the access to use this API.

Sample Request: Querying all user instances request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
view_registration_requests
Accept: Authorization

Content-Type: Text

Example for querying all users:

```
eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJCmVZb0b2t1biA6IHRva2VuX1Z0Rk0iLCJpYXQiOiE1NzAwMDYsInN1YiI6I1N1YmplY3Qgb2YgSldUIiwiaXNzIjoisXNzdWVyIG9mIFRva2VuOiBPcmFjbGUTRFNSIiwiaXVkiIjoisYXVkaWVuY2UgOiBhZG1pbiIsIm4cCI6MTU3MzAzOTIwNn0.Ep-lKGBZqa09u_cpj1bSN8DBpwvZoRMQTOYNr18KY8w
```

Sample Response: Querying all user instances Response

201 Created
Content-Type: application/json

Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
view_registration_requests

```
{
  "user": [
    {
      "username": "xx",
      "password": "xx",
      "access": "read"
    },
    {
```

```
"username": "xx",
"password": "xx",
"access": "read"
},
{
"username": "xx",
"password": "xx",
"access": "admin"
}
]
```

The following table describes the parameters used for sending request to VNFM:

Table 8-3 Parameters and Definitions

Parameter	Definitions
Authorization	Authentication Token generated by the admin.

8.5 Change Password of the User

The user must provide the username, old password, new password and the Authorization to change the existing credentials in the system.

Note

A valid password must be in range between 8 to 31 characters, with at least one digit, at least one lowercase letter, at least one uppercase letter, at least one special character, and should not contain white spaces.

Sample Request: Change Password request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/change_password

Accept: application/json

Content-Type: application/json

Example for changing the password for a user:

Authorization :

eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJRCBvZiB0b2t1biA6IHRva2VuXzI0RkoiLCJpYXQiOjE1NzMwMjEyMDY5InN1YiI6IlN1YmplY3Qgb2YgSldUIiwiaXNziJjoisXNzdWVyIG9mIFRva2VuOiBPcmFjbGUtRFNSIiwiaXVkaWJoiYXVkaWVudUg0bHVGZG1pb2IsIm4cCI6MTU3MzAzOTIwNn0.Ep-lKGBZqa09u_cpj1bSN8DBpwvZoRMQT0YNr18KY8w

 $\{$

```
"username": "xxx",  
"newPassword": "xxx",  
"oldPassword": "xxx"  
}
```

Sample Response: Change Password Response

201 Created

Content-Type: application/json

Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
change_password

```
{  
  
"response": "Password Updated Successfully."  
  
}
```

The following table describes the parameters used for sending request to VNFM:

Table 8-4 Parameters and Definitions

Parameter	Definitions
username	Username of the existing user.
newPassword	The new password to be set for the user.
oldPassword	The already existing password for the user.

8.6 Provision and Add User to VNFM

The admin must provide the username, password, access grant and the Authorization to add the credentials in the system.

Note

Only the admin has the access to use this API.

Sample Request: Provision and Adding the users request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/provision_and_add

Accept: application/json

Content-Type: application/json

Example for provisioning and adding a user:

Authorization :

```
eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJRCBvZiB0b2t1biA6IHRva2VuX1Z0Rk0iLCJpYXQiOiE1NzYyMDYsInN1YiI6I1N1YmplY3Qgb2YgSldUIiwiaXNzIjoisXNzdWVyaW9mIFRva2VuOiBPcmFjbGUtRFNSIiwiaXVkaWVudUg0IjBhZG1pbIsIm4cCI6MTU3MzAzOTIwNn0.Ep-lKGBZqa09u_cpj1bSN8DBpwVZoRMQTOYNr18KY8w
```

```
{
  "username": "xxx",
  "password": "xxx",
  "access": "read/admin"
}
```

Sample Response: Provision and Adding Response

201 Created

Content-Type: application/json

Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/provision_and_add

```
{
  "response": "User Successfully created."
}
```

The following table describes the parameters used for sending request to VNFM:

Table 8-5 Parameters and Definitions

Parameter	Definitions
username	Username of the user.
password	Password of the user.
access	Scope of the user.

9

VNFM State Information

VNFM State Information consists of the following:

- State of the VNFM
- IP address of Primary VNFM
- IP address of Secondary VNFM
- Last updated TimeStamp

Query VNFM State Information

The State of VNFM can be queried using the REST GET request.

Sample Request: Query VNFM State Info

URL: GET: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnfmState`

Authorization : <Token generated after login>

Sample Request: Response for Query VNFM State Info

```
{
  "state": "PRIMARY",
  "primaryIp": "2606:b400:605:b813::7",
  "secondaryIp": "2606:b400:605:b813::4",
  "timeStamp": "2020/07/02 20:29:08 UTC"
}
```

9.1 Change VNFM State

The VNFM state can be changed using the CHANGE VNFM STATE' POST REST request.

Sample Request: Change VNFM Request

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnfmState`

Accept: `application/json`

Content-Type: `application/json`

Authorization : <Token generated after login>

```
{
```

```
"state": "SECONDARY",  
  
"primaryIp": "2606:b400:605:b813::4",  
  
"secondaryIp": "2606:b400:605:b813::7"  
  
}
```

Sample Request: Responses for Change VNFM State

201 Created

```
"localStateInfo": {  
  
  "state": "SECONDARY",  
  
  "primaryIp": "2606:b400:605:b813::4",  
  
  "secondaryIp": "2606:b400:605:b813::7"  
  
},  
  
"remoteStateInfo": {  
  
  "state": "PRIMARY",  
  
  "primaryIp": "2606:b400:605:b813::7",  
  
  "secondaryIp": "2606:b400:605:b813::4"  
  
},  
  
"timeStamp": "2020/07/03 10:35:29 UTC"  
  
}
```

202 Accepted

```
{  
  
  "localStateInfo": {  
  
    "state": "SECONDARY",  
  
    "primaryIp": "2606:b400:605:b813::4",  
  
    "secondaryIp": "2606:b400:605:b813::7"  
  
  },  
  
  "remoteStateInfo": null,  
  
  "timeStamp": "2020/07/03 10:35:29 UTC"  
  
}
```

Note

- The 201 response indicates that the request is complete.
- The 202 response indicates that the request is accepted for processing. The VNF retries to update the state for the configured amount of time. Use the LCM operation ID to determine the status of the request.
 - If the request is complete, then the LCM operation status changes to COMPLETED.
 - If the request fails, then the VNFM state changes to TRANSIENT and the LCM operation status changes to FAILED. In this case, the user must re-trigger the CHANGE VNFM request after resolving the reason for failure.

9.1.1 VNFM State Parameter Description

Table 9-1 VNFM State Parameter Description

Parameter	Description
state	New VNFM state to be changed to. Only two values are accepted: PRIMARY and SECONDARY.
primaryIp	The new Primary VNFM IP.
secondaryIp	The new Secondary VNFM IP. It can be updated to 127.0.0.1 to make it a standalone VNFM.

9.2 Making VNFM as a Standalone VNFM

If there is a permanent loss of a VNFM, then the model can be made into a standalone system using the CHANGE VNFM STATE REST request.

Sample Request: Request for Standalone VNFM

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnfmState`

Accept: `application/json`

Content-Type: `application/json`

Authorization : `<Token generated after login>`

```
{
  "state": "SECONDARY",
  "primaryIp": "2606:b400:605:b813::4",
  "secondaryIp": "127.0.0.1"
}
```

Sample Request: Response for Change VNFM Request Standalone VNFM

201 Created

```
{
  "localStateInfo": {
    "state": "PRIMARY",
    "primaryIp": "2606:b400:605:b813::4",
    "secondaryIp": "127.0.0.1"
  },
  "remoteStateInfo": {
    "message": "Remote IP does not exist or is LocalHost. Remote Ip:
127.0.0.1"
  },
  "timeStamp": "2020/07/03 14:43:31 UTC"
}
```

Steps to Change the Default Retry Configurations

When Change VNFM State Info Request gives a 202 response, a retry mechanism occurs to change the VNFM state. By default, the retry occurs for 10 times with a 2 min wait interval between each retry. These configurations can be changed with the following steps.

In 'dsrvnfm' user mode:

1. Open VnfmProperties file /opt/vnfm/config/VnfmProperties.yaml to edit the properties.
2. Retry Count can be modified from the following property changeRemoteStateRetryCount.
 - a. Min Value: 2
 - b. Max Value: 10
 - c. Default Value: 10
3. Retry Interval can be modified from the following property changeRemoteStateRetryInterval. The interval is in the form of milliseconds.
 - a. Min Value: 60000
 - b. Max Value: 300000
 - c. Default Value: 120000

9.3 Changing the Default Retry Configuration

When Change VNFM State Info Request gives a 202 response, a retry mechanism starts to change the VNFM state. By default, the retry occurs 10 times with a 2 minute wait interval between each retrial. Perform the following procedure to change this configuration:

1. As a dsrvnfm user, open the /opt/vnfm/config/VnfmProperties.xml VnfmProperties file to edit the properties.
2. Change the retry count value using changeRemoteStateRetryCount.
 - Min Value: 2
 - Max Value: 10
 - Default Value: 10
3. Change the retry interval value using changeRemoteStateRetryInterval.
 - Min Value: 60000
 - Max Value: 300000
 - Default Value: 120000

The retry interval value is in the unit of milliseconds.

9.4 API Versions

- VNFM supports the following dedicated URIs to enable API consumers to retrieve information about API versions supported.
- For the VNF lifecycle management interface version as specified in the present document, the MAJOR version field shall be 2, the MINOR version field shall be 2, and the PATCH version field shall be 0.
- API version identifiers shall consist of 3 numerical fields, following a MAJOR.MINOR.PATCH.

For Common API Version

Sample Request:

Query Common API Versions

URL: GET: https://<VNFM HOST IP>:8443/docs/vnfm/
Authorization : <Token generated after login>

Sample Response:

Response for Query Common API Versions

```
{
  "uriPrefix": "https://10.75.210.41:8443/vnflcm/api_versions",
  "apiVersions": [
    {
      "version": "2.2.0",
      "isDeprecated": false
    },
    {
      "version": "1.1.0",
      "isDeprecated": false
    }
  ]
}
```

For V1 API Version

Sample Request:

Response for Query V1 API Version

URL: GET: https://<<VNFM HOST IP>>:8443/docs/vnfm/v1/

Authorization : <Token generated after login>

Sample Response

Response for Query V1 API Version

```
{
  "uriPrefix":
    "https://10.75.210.41:8443/vnflcm/v1/api_versions",
  "apiVersions": [
    { "version": "1.1.0", "isDeprecated": false
  }
]
```

For V2 API Version

Sample Request:

Query V2 API Version

URL: GET: https://<<VNFM HOST IP>>:8443/docs/vnfm/v2

Authorization : <Token generated after login>

Sample Response:

Response for Query V2 API Version

```
{
  "uriPrefix": "https://10.75.210.41:8443/vnflcm/v2/api_versions",
  "apiVersions": [
    {
      "version": "2.2.0",
      "isDeprecated": false
    }
  ]
}
```

Parameters and Definitions

Table 9-2 Parameters and Definitions

Parameters	Definitions
uriPrefix	Specifies the prefix of the resource URI of the "API versions" resource represented by this data structure. Depending on the resource URI, it shall be in one of the two following forms: {apiRoot}/{apiName}/{apiMajorVersion}/ or {apiRoot}/{apiName}/
apiVersions	Versions supported for the API signalled by the uriPrefix attribute.
version	Identifies a supported version.
isDeprecated	If such information is available, this attribute indicates whether use of the version signalled by the version attribute is deprecated (true) or not (false).

10

Deploying VNFs

Prerequisites: A virtual infrastructure satisfying the [DSR VNFM OpenStack Prerequisites](#).

Table 10-1 Supported VNFM Network Interfaces

Node Type	IPv4	Multiple XSI	Fixed XMI	Fixed XSI/SBR	Fixed IMI	IPv6 XSI	IPv6 XMI	IPv6 IMI	Cloud-init
DSR									
DSR NOAM	Y	NA	Y	NA	Y	NA	Y	Y	Y
DR DSR NOAM	Y	NA	Y	NA	Y	NA	Y	Y	Y
DSR SOAM	Y	NA	Y	NA	Y	NA	Y	Y	Y
DAMP	Y	Y	Y	Y	Y	Y	Y	Y	Y
vSTP MP	Y	Y	Y	Y	Y	Y	Y	Y	Y
SERVICE MP	Y	NA	Y	NA	Y	NA	Y	Y	Y
IPFE	Y	Y	Y	Y	Y	Y	Y	Y	Y
IDIH	Y	NA	Y	NA	Y	NA	N	N	Y
SBR	Y	NA	Y	Y (SBR Replication Ports)	Y	NA	Y	Y	PARTIAL *
UDR NOAM	Y	Y	Y	Y	Y	Y	Y	Y	Y
SDS									
SDS NOAM	Y	NA	Y	NA	Y	NA	Y	Y	Y
Query Server	Y	NA	Y	NA	Y	NA	Y	Y	Y
DR SDS NOAM	Y	NA	Y	NA	Y	NA	Y	Y	Y
SDS SOAM	Y	NA	Y	NA	Y	NA	Y	Y	Y
DP Server	Y	NA	Y	NA	Y	NA	Y	Y	Y
ATS									
ATS MASTER	Y	Y(2)	Y	Y	NA	Y	Y	NA	NA

Signaling flavors with DIAMETER+STP with multiple xsi (2 XSI interface)

Partial*: Cloud init for SBR servers are not supported completely.

- The servers are added as plain SBRs (Not as Session, Binding or Universal).
- The server groups are created according to the flavor. (Check [Table 4-1](#) for more information)
- Depending on the flavor, there will be a SBR left out from the server group.
- The left out server group should be added to the mated site's server group manually.
- VNFM does not perform the PDRA configuration. Users have to configure PDRA manually.

10.1 Create a VNF Instance

1. Before a DSR VNF is instantiated, the user must first issue a request to create a VNF instance by using the command **create VNF instance**.
2. Creating a VNF instance informs the VNFM that a user has requested to instantiate a VNF at some point in the future.
3. The VNFM returns a VNF ID that must be saved for future use while performing operations on the same VNF.

Note

- Each VNF has its own VNF ID, so if it is required to create a DSR with two signaling VNFs, then issue the request to create a VNF instance three times, once for the network OAM VNF, and once for each signaling VNFs.
- The `vnfInstanceName` value is defined as per the following:
 - It is provided as the prefix of the VMName / Hostname for each VNFc in any VNF. It is an optional parameter, if not provided, then a default value is generated.
 - The `vnfInstanceName` includes only alphanumeric characters, and special character such as ' - ' (Hyphen). It must start with an alphabet. No other special character except ' - ' (Hyphen) is allowed.
 - Max allowed length is 22 characters.

For more information about the full list of all inputs and possible outputs of the **create VNF instance** command, see **ETSI NFV-SOL 003**, section **5.4.2.3.1**, or the DSR VNFM Swagger specification.

VNFM Multiple VNFD ID Support

Currently, VNFM supports only fixed VNFD ID for each VNF type. The customer needs to deploy multiple DSR projects/VNF's using a single NFVO.

VNFD ID is used as a key parameter to identify and deploy the VNF using NFVO. To deploy multiple DSR VNF using NFVO, the customer needs the flexibility to use multiple VNFD ID.

Table 10-2 VNFM Multiple VNFD ID support (VNFM-V1)

Req No	Description
R-33473666-100	VNFM shall continue to support existing VNFD ID fields for backward compatibility.
R-33473666-200	VNFM shall allow suffix of underscore and max 3 characters to VNFD ID in the following format. e.g. <i>dsrNetworkOam_Xyz</i>

Table 10-2 (Cont.) VNFM Multiple VNFD ID support (VNFM-V1)

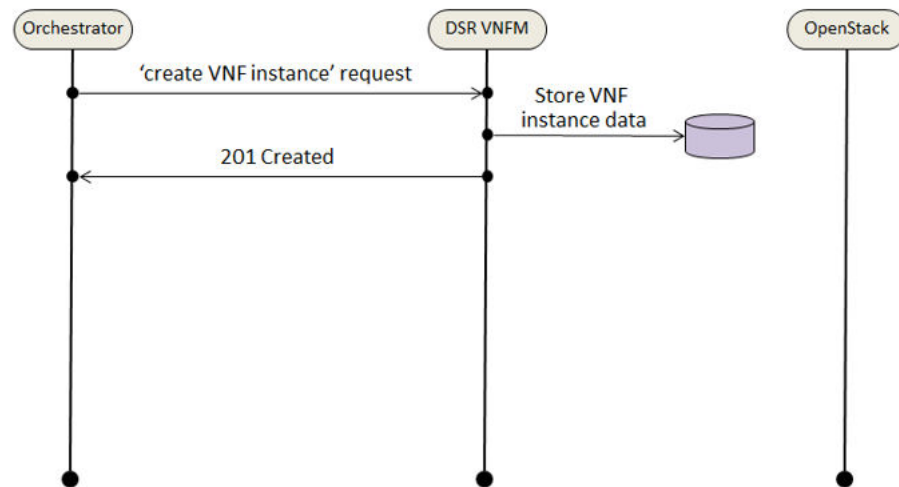
Req No	Description
R-33473666-300	Suffix shall support Alpha-numeric only. e.g "vnfdId": "dsrNetworkOam_ABC" "vnfdId": "dsrNetworkOam_abc" "vnfdId": "dsrNetworkOam_Abc" "vnfdId": "dsrNetworkOam_A1B" "vnfdId": "dsrNetworkOam_a1y" "vnfdId": "dsrNetworkOam_123"
R-33473666-400	VNFM shall allow suffix for following VNF deployments 1. NOAM 2. DR NOAM 3. SOAM and Signalling 4. SDS NOAM 5. SDS DR NOAM 6. SDS Signaling
R-33473666-500	Adding a suffix to VNFD ID shall not impact the existing support LCM operations. i.e. Create, Instantiate, scale, heal, Query, get operations status etc.
R-33473666-600	To support this feature customers will have to deploy the new build of VNFM. Upgrade of VNFM will not be supported.

Query Individual VNF Package (VNFM - V2)

Precondition: The VNF package is on-boarded to the NFVO.

- This API will be triggered by VNFM internally when Orchestrator triggers Create VNF API operation.
- The VNFM sends a GET request to the "VNFD in an individual VNF package" resource to NFVO.
- The NFVO returns a "200 OK" response and includes a copy of the VNFD from the VNF package in the payload body to the VNFM.

The following image illustrates the VNF instance creation:

Figure 10-1 VNF Create Instance Request

Sample Request: Create VNF instance request generated.

Resource URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances`
 Accept: `application/json`
 Content-Type: `application/json`
 Authorization: Token generated after login

Create VNF instance request generated (VNFM - V1)

Example for VNFM Secondary:

```
{
  "vnfdId": "vnfmSecondary",
  "vnfInstanceName": "DemoVnfmSecondary",
  "vnfInstanceDescription": "DemoVnfmSecondaryDescription"
}
```

Example for NOAM:

```
{
  "vnfdId": "dsrNetworkOam",
  "vnfInstanceName": "DemoNoam",
  "vnfInstanceDescription": "DemoNoam "
}
```

Example for **DR NOAM**:

```
{
  "vnfdId": "dsrDrNetworkOam",
  "vnfInstanceName": "DemoDrNoam",
  "vnfInstanceDescription": "DemoDrNoam "
}
```

Example for **Signaling**:

```
{
  "vnfdId": "dsrSignaling",
  "vnfInstanceName": "DemoSoam",
  "vnfInstanceDescription": "Description"
}
```

Example for **SDS NOAM**

```
{
  "vnfdId": "sdsNetworkOam",
  "vnfInstanceName": "DemoSdsNoam",
  "vnfInstanceDescription": "DemoSdsNoam "
}
```

Example for **SDS DR NOAM**:

```
{
  "vnfdId": "sdsDrNetworkOam",
  "vnfInstanceName": "DemoSdsDrNoam",
  "vnfInstanceDescription": "DemoSdsDrNoam "
}
```

Example for **SDS Signaling**:

```
{
  "vnfdId": "sdsSignaling",
  "vnfInstanceName": "DemoSdsSoam",
  "vnfInstanceDescription": "DemoSdsSignaling"
}
```

Example for **ATS Master**:

```
{
  "vnfdId": "atsMaster",
  "vnfInstanceName": "DemoAtsMaster",
  "vnfInstanceDescription": "DemoAtsMaster"
}
```

Example for **IDIH**:

```
{
  "vnfdId": "dsrIdih",
```

```

    "vnfInstanceName": "DemoIdih",
    "vnfInstanceDescription": "Description for IDIH VNF"
  }

```

Sample Response

201 Created

Create VNF Instance Response

Content-Type: application/jsonFull Semantic Validation Configuration

Authorization: Token generated after login

Resource URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances

Create VNF Instance Response (VNFM - V1)

```

{
  "id": "dsrNetwork0am-b44e9a45-b575-4b30-b580-085d8ddd7015",
  "vnfdId": "dsrNetwork0am",
  "instantiationState": "NOT_INSTANTIATED",
  "vnfInstanceName": "DemoNoam",
  "vnfInstanceDescription": "string",
  "vnfProvider": "Oracle",
  "vnfProductName": "DSR",
  "vnfSoftwareVersion": "DSR-9.0.2.0.0",
  "vnfdVersion": "5.x",
  "onboardedVnfPkgInfoId": "N/A",
  "links": {
    "self": {
      "href": "https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-793a2420-adab-4347-9667-489ae671b767"
    },
    "instantiate": {
      "href": "https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-793a2420-adab-4347-9667-489ae671b767/instantiate"
    },
    "scaleToLevel": null,
    "terminate": null
  }
}

```

Note

VNFM supports both the secured and the unsecured URL (HTTPS with port 8443 and HTTP with port 8080).

The following table describes the parameters used for sending request to VNFM:

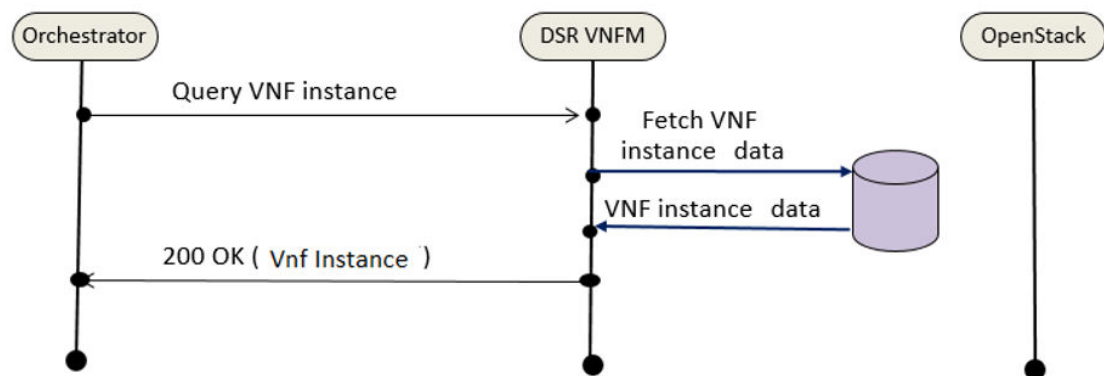
Table 10-3 Parameters and Definitions for VNF Instance V1

Parameter	Definition
vnfdId	Identifier of the VNF instance deployment ID to be created
vnfInstanceName (optional)	Name of the VNF instance to be created (must be unique)
vnfInstanceDescription (optional)	Description of the VNF instance. T

10.2 Query VNF Instance

The diagram describes a sequence for querying/reading information about a VNF instance.

< Below Diagram and Associated Note - TBD >

Figure 10-2 Query VNF Instance

VNF instance query, as illustrated above, performs the following actions:

- If the NFVO intends to read information about a particular VNF instance, it sends a GET request to the **Individual VNF instance** resource, addressed by the appropriate VNF instance identifier (Vnf Id) in its resource URI.
- The VNFM returns a **200 OK** response to the NFVO, and includes specific data structure of type **VnfInstance** related to the VNF instance identifier (Vnf Id) in the payload body.
- If the NFVO intends to query all VNF instances, it sends a GET request to the **VNF instances** resource.
- The VNFM returns a **200 OK** response to the NFVO, and includes zero or more data structures of type **VnfInstance** in the payload body.

10.2.1 Query Individual VNF Instance

Sample Request for Single VNF Instance:

Querying VNF Instances

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<<VNF Instance Id>>

Authorization : <Token generated after login>

Sample Response for Query Individual VNF Instances:

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<<VNF Instance Id>>

Accept: application/json

Content-Type: application/json

Query Individual Response:

Query Individual VNF Instance (VNFM - V1)

```
{
  "id": "dsrNetwork0am-c689e44d-2b93-473f-935a-3bf09957fe9f",
  "vnfdId": "dsrNetwork0am",
  "instantiationState": "INSTANTIATED",
  "vnfInstanceName": "dsrvnfm",
  "vnfInstanceDescription": "dsrvnfm",
  "vnfProvider": "Oracle",
  "vnfProductName": "DSR",
  "vnfSoftwareVersion": "DSR-9.0.2.0.0",
  "vnfdVersion": "5.x",
  "onboardedVnfPkgInfoId": "N/A",
  "links": {
    "self": {
      "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-c689e44d-2b93-473f-935a-3bf09957fe9f"
    },
    "instantiate": {
      "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-c689e44d-2b93-473f-935a-3bf09957fe9f/instantiate"
    },
    "scaleToLevel": {
      "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-c689e44d-2b93-473f-935a-3bf09957fe9f/scale_to_level"
    },
    "terminate": {
      "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-c689e44d-2b93-473f-935a-3bf09957fe9f/terminate"
    }
  },
  "instantiatedVnfInfo": {
    "flavourId": "DSR NOAM",
    "vnfState": "STARTED",
    "extCpInfo": {
      "id": null,
      "cpdId": null
    },
    "scaleStatus": [{
      "aspectId": "NOAM",
      "scaleLevel": "2"
    }
  ]
}
```

```

    ]]
  },
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  }
}
}
}

```

10.2.2 Query All VNF Instances (VNFM-V1)

Sample Request for all VNF Instances:

Querying VNF Instances

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances

Authorization: <Token generated after login>

Sample Response for Query All VNF Instances :

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances

Accept: application/json

Content-Type: application/json

Response Body for No VNF Instances []

```
[ ]
```

Response Body for Query All VNF Instances:

Query All VNF Instances (VNFM - V1)

```

[
  {
    "id": "dsrNetwork0am-38f694dc-be36-4747-814d-5fccd4fa6163",
    "vnfdId": "dsrNetwork0am",
    "instantiationState": "INSTANTIATED",
    "vnfInstanceName": "string",
    "vnfInstanceDescription": "dsrvnfm",
    "vnfProvider": "Oracle",
    "vnfProductName": "DSR",
    "vnfSoftwareVersion": "DSR-9.0.2.0.0",
    "vnfdVersion": "5.x",
    "onboardedVnfPkgInfoId": "N/A",
  }
]

```

```

    "links": {
      "self": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
vnf_instances/dsrNetwork0am-38f694dc-be36-4747-814d-5fccd4fa6163"
      },
      "instantiate": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
vnf_instances/dsrNetwork0am-38f694dc-be36-4747-814d-5fccd4fa6163/instantiate"
      },
      "scaleToLevel": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
dsrNetwork0am-38f694dc-be36-4747-814d-5fccd4fa6163/scale_to_level"
      },
      "terminate": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
dsrNetwork0am-38f694dc-be36-4747-814d-5fccd4fa6163/terminate"
      }
    },
    "instantiatedVnfInfo": {
      "flavourId": "DSR NOAM",
      "vnfState": "STARTED",
      "extCpInfo": {
        "id": null,
        "cpdId": null
      },
      "scaleStatus": [
        {
          "aspectId": "NOAM",
          "scaleLevel": "2"
        }
      ]
    },
    "vimConnectionInfo": {
      "Mv1Cloud": {
        "vimid": "vimid",
        "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
        "interfaceInfo": {
          "endpoint": "https://mv1-dev1.us.oracle.com:5000/v3"
        }
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  {
    "id": "dsrNetwork0am-31fd9dc5-bcce-4dfb-ae21-46f07cd3cba5",
    "vnfdId": "dsrNetwork0am",
    "instantiationState": "NOT_INSTANTIATED",
    "vnfInstanceName": "demo",
    "vnfInstanceDescription": "dsrvnfm",

```

```

    "vnfProvider": "Oracle",
    "vnfProductName": "DSR",
    "vnfSoftwareVersion": "DSR-9.0.2.0.0",
    "vnfdVersion": "5.x",
    "onboardedVnfPkgInfoId": "N/A",
    "links": {
      "self": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
vnf_instances/dsrNetwork0am-31fd9dc5-bcce-4dfb-ae21-46f07cd3cba5"
      },
      "instantiate": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
vnf_instances/dsrNetwork0am-31fd9dc5-bcce-4dfb-ae21-46f07cd3cba5/instantiate"
      },
      "scaleToLevel": null,
      "terminate": null
    }
  }
}

```

10.2.3 Query all VNF Instances Using Filter

VNFM Supports query with filter option where user need to pass filter data in specific format. We support operations like **eq**, **in**, **nin**, **cont** and **ncont**.

Format: (**operation**, **field**, **value**)

Example: (**eq**, **vnfProductName**, **DSR**)

Sample Request for all VNF Instances with filter:

Querying VNF Instances

URL: GET: https://<>:8443/vnflcm/<>/vnf_instances

Authorization : <Token generated after login>

Filter: (eq, vnfInstanceName, FT--da8d7130)

Sample Response for Query All VNF Instances Using Filter :

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances

Accept: application/json

Content-Type: application/json

Query All Using filter

```

[ {
  "id": "dsrNetwork0am_DRA-1f57534f-18f7-487f-b934-f8298958b796",
  "vnfdId": "dsrNetwork0am_DRA",
  "instantiationState": "INSTANTIATED",
  "vnfInstanceName": "FT--da8d7130",
  "vnfInstanceDescription": "dsrvnfm",
  "vnfProvider": "Oracle", "vnfProductName": "DSR", "vnfdVersion":
"6.0.2",
  "links": {
    "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/
vnf_instances

```

```

        /dsrNetworkOam_DRA-1f57534f-18f7-487f-b934-f8298958b796"
    },
    "instantiate": {
        "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/
vnflcm/v1/vnf_instances
        /dsrNetworkOam_DRA-1f57534f-18f7-487f-b934-f8298958b796"
    },
    "scaleToLevel": {
        "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/
vnflcm/v1/vnf_instances
        /dsrNetworkOam_DRA-1f57534f-18f7-487f-b934-f8298958b796"
    },
    "operate": { "href":
        "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/
vnf_instances
        /dsrNetworkOam_DRA-1f57534f-18f7-487f-b934-f8298958b796"
    },
    "terminate": { "href":
        "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/
vnf_instances
        /dsrNetworkOam_DRA-1f57534f-18f7-487f-b934-f8298958b796"
    },
    "heal": { "href":
        "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/
vnf_instances
        /dsrNetworkOam_DRA-1f57534f-18f7-487f-b934-f8298958b796" }
    },
    "additionalInfo": {
        "dsrNoam": [ { "FT--da8d7130-DN000": { "image":
"DSR-9.0.2.0.0_98.4.0",
        "status": "STARTED", "serverId": "4afcbda2-1563-4bcb-8f2f-
a3ec208ca484" }
        },
        {
            "FT--da8d7130-DN001": { "image": "DSR-9.0.2.0.0_98.4.0",
"status": "STARTED",
            "serverId": "f31d36ba-26a0-400e-af91-2179560e7594"
        }
        }
    ],
    "instantiatedVnfInfo": { "flavourId": "DSR NOAM", "vnfState": "STARTED",
        "extCpInfo": null, "scaleStatus": [ { "aspectId": "NOAM",
"scaleLevel":
        "2" } ]
    },
    "vimConnectionInfo": {
        "MvlCloud": { "vimId": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
            "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        }
    },
    "accessInfo": {
        "username": "dsrvnmf",
        "password": "YXV0b21hdGlvbg==",
        "userDomain": "default",
        "projectDomain": "default",
        "project": "VNFM_FT1", "region": "VNFM_FT1" },
        "extra": {}
    }
}

```

```

    }
  }
}
]

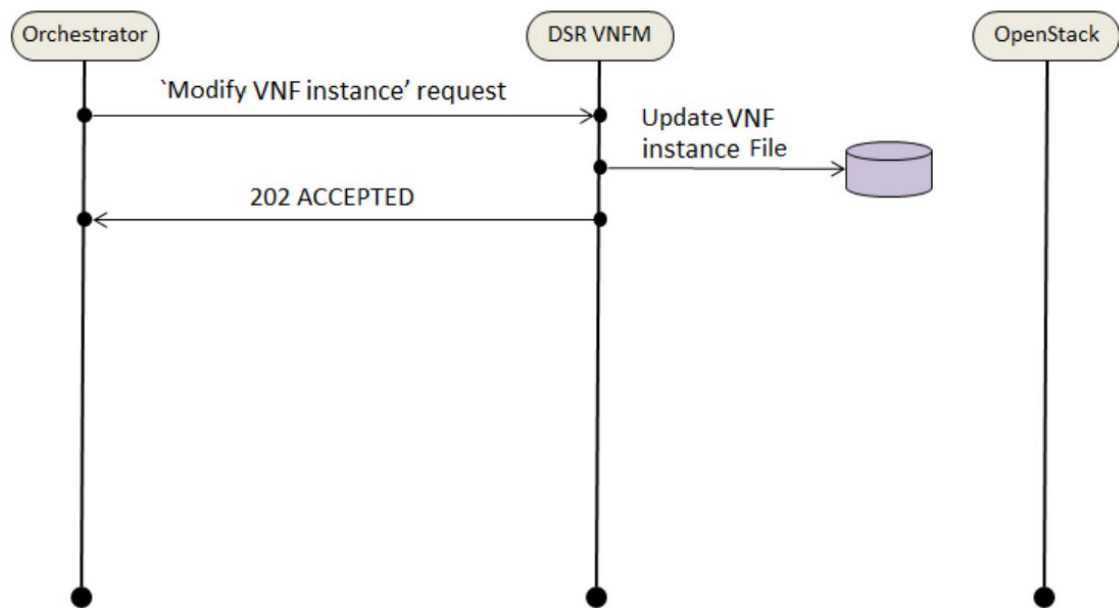
```

10.3 Modify VNF Information Operation

VNFM supports the API for "Modify VNF operation". So the VNF Identifier resources created can be modified .

The modifiable fields are "vnfInstanceDescription" and "vimConnectionInfo".

Figure 10-3 Modify VNF Instance Request



Precondition: The resource representing the VNF instance has been created that indicates the state must be in the INSTANTIATED state.

Postcondition: VNF LCM operation displays the status of this operation and updates the VNF instance file at the end.

Sample Request for Modify VNF

Modify VNF

Request URL : PATCH:URL: `https://<VNFM HOST IP>:8443/vnfm/<vnfm_version>/vnf_instances/< VNF ID received from instantiate request>`

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
  "vnfInstanceDescription": "Modify VNF Information Data",
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",

```

```

    "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
    "interfaceInfo": {
      "endpoint": "https://mv1-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default"
    }
  }
}

```

Note

The 202 response indicates that the request is accepted for processing. Use Query LCM operation for status.

Response of Modify VNF

202 Accepted

Headers:

```

{
  cache-control: private
  connection: keep-alive
  content-length: 0
  date: Mon, 06 Mar 2023 10:19:08 GMT
  keep-alive: timeout=60
  location: https://localhost:7777/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
    lcmOp-2b7bb209-53ea-422c-b3f7-4d1dd4fcb155
  strict-transport-security: max-age=31536000;includeSubDomains;preload
  x-content-type-options: nosniff
  x-frame-options: DENY
  x-xss-protection: 1; mode=block
}

```

Table 10-4 Parameters and Definitions for Modify VNF

Parameter	Definition
vnfInstanceDescription (Optional)	Updates vnfInstanceDescription details.
vimConnectionInfo (Optional)	Updates vimConnectionInfo details.

Note

You must provide one of the mentioned parameters in the request body to activate Modify VNF API.

10.4 Deleting a VNF Instance

VNFM supports the LCM function of "Delete VNF identifier". So that the VNF Identifier resources created are deleted.

Precondition: The resource representing the VNF instance to be deleted needs to be in NOT_INSTANTIATED state.

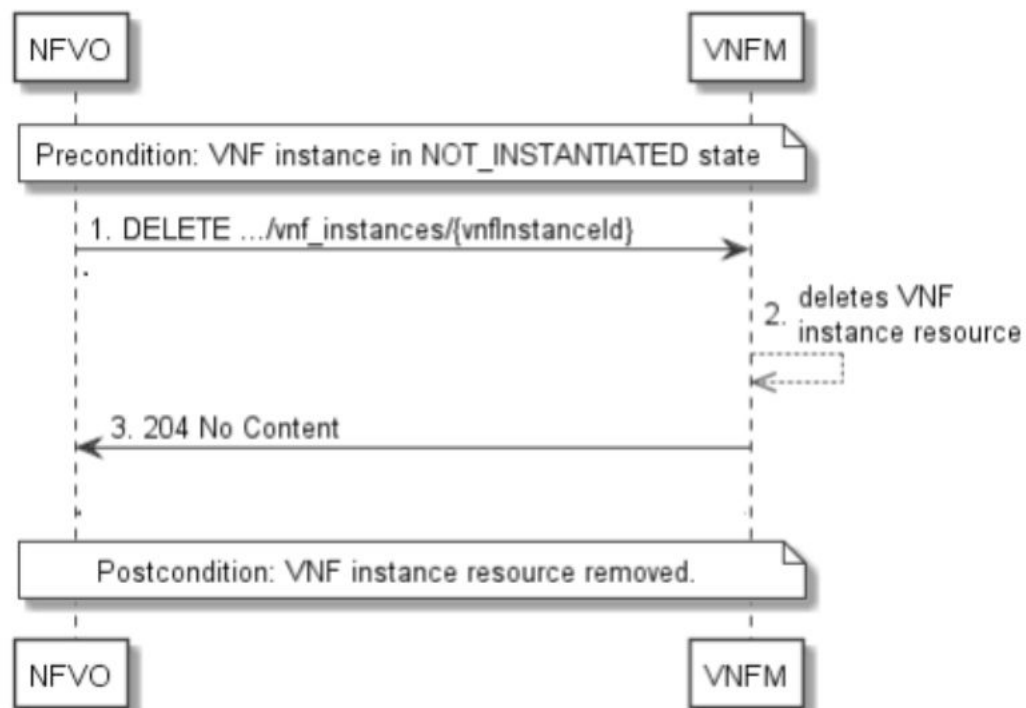
Deletion of a VNF Instance happens in the following sequence:

- NFVO sends a DELETE request to the "Individual VNF Instance" resource.
- The VNFM deletes the VNF instance resource and the associated VNF instance identifier.
- The VNFM returns a "204 No Content" response with an empty payload body.

Result: The resource representing the VNF instance has been removed from the list of VNF instance resources.

The following diagram describes the flow of deletion of a VNF Instance Resource.

Figure 10-4 Deleting a VNF Instance Resource



DELETE Operation

URL: DELETE: https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/<<{vnfInstanceId}>>

DELETE Operation

Response Code : 204

```
{
  cache-control: private connection: keep-alive
  date: Mon, 06 Mar 2023 10:14:37 GMT
  keep-alive: timeout=60
  strict-transport-security:max-age=31536000;includeSubDomains;preload
  x-content-type-options: nosniff
  x-frame-options:DENY
  x-xss-protection: 1; mode=block
}
```

10.5 Grant Lifecycle Operation (VNFM - V2)

- This operation allows the VNFM to obtain from the NFVO permission and configuration parameters for a VNF lifecycle operation. The NFVO can approve or reject a request based on policies, for example, dependencies between VNFs and available capacity.
- If Grant failed, respective operation returns LCM status as FAILED.
- This API is invoked internally from Instantiate, Scale to level, Terminate, Heal and Operate VNF API.

10.6 Instantiating VNFM Secondary

VNFM Secondary is a Geo Redundant VNFM. Instantiating a Secondary VNFM is supported if there is no secondary VNFM configured in the Primary VNFM. Primary VNFM must be a standalone VNFM.

Both dynamic and fixed IP deployments are supported. Before deploying the VNFM, the following information must be available:

- The VNF ID for a previously created VNFM Secondary VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
 - OpenStack endpoint
 - User Domain Name
 - Project Domain ID
 - User name
 - Password

- Project
- The name of a public network in the selected OpenStack instance that carries the VNFM traffic.
- The name of a network in the selected OpenStack instance that carries the VIM traffic.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server.

For more information about the list of inputs and possible outputs of the command `instantiate VNF`, refer to ETSI NFV-SOL 003 or the DSR VNFM Swagger.

10.6.1 Sample Request for DYNAMIC IP Model

Dynamic deployment model for VNFM Secondary

Request DYNAMIC IP Model (VNFM - V1)

```
{
  "flavourId": "VNFM SECONDARY",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],
  "vimConnectionInfo": [
    {
      "MvlCloud": {
        "vimid": "vimid",
        "vimType": "\"ETSINFV.OPENSTACK_KEYSTONE.V_3\"",
        "interfaceInfo": {
          "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        }
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default",
      }
    }
  ],
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "vnfmNetwork": {
      "name": "ext-net-ipv6",
      "ipVersion": "IPv6",
      "subnet": "ext-net-ipv6-subnet"
    }
  }
}
```

```

    },
    "vimNetwork": {
        "name": "ext-net",
        "ipVersion": "IPv4",
        "subnet": "ext-net-subnet"
    },
    "ntpServerIp": "10.250.32.10",
    "secondaryVnfmVolumeId": "174c7cae-d9f0-4459-9463-a0c20443ef0c",
    "vimRouteAddress": "10.75.167.0/24,10.75.185.0/24,10.75.171.192/26",
    "image": "https://gbuconfluence.oraclecorp.com/pages/editpage.action?
pageId=167347102",
    "flavor": "vnfm",
    "availabilityZone": "nova"
}
}

```

vnfConfigurableProperties Json:

DYNAMIC IP Model (VNFM - V2)

```

{
    "ntpServerIp" : "2606:b400:605:b912:200:5eff:fe00:1f7",
    "vimNetwork" : {
        "name" : "ext-net-lab",
        "ipVersion" : "IPv6",
        "subnet" : "ext-net-lab-subnet"
    },
    "secondaryVnfmVolumeId" : "5490fb59-e1e4-4435-8510-4d22b7772d04",
    "vimRouteAddress" : "2606:b400:605:b83b::/
64,2606:b400:605:b857::/64,2606:b400:605:b854::/64,2606:b400:605:b849::/64,fd0
d:deba:d97c:3f::/64,2606:b400:605:b83d::/64" }

```

10.6.2 Sample Request for FIXED IP Model

Request FIXED IP Model (VNFM - V1)

```

{
    "flavourId": "VNFM SECONDARY",

```

```

"instantiationLevelId": "HA",

"extVirtualLinks": "extVirtualLinks",

    "extManagedVirtualLinks": [],

"vimConnectionInfo": {
  "MvlCloud": {
    "vimId": "vimid",

    "vimType": "\"ETSINFV.OPENSTACK_KEYSTONE.V_3\"",

    "interfaceInfo": {
      "endpoint": "\"https://mvl-dev1.us.oracle.com:5000/v3\""
    },

    "accessInfo": {
      "username": "dsrvnfm",

      "password": "xxxxxx",

      "userDomain": "Default",

      "projectDomain": "Default",

      "tenant": "VNFM_FT1"
    }
  },

  },

"localizationLanguage": "localizationLanguage",

"additionalParams": {
  "vnfmNetwork": {
    "name": "ext6-net-4",

    "ipVersion": "IPv6",

    "subnet": "ext6-subnet-4",

    "fixedIps": {
      "vnfmIp": "2606:b400:605:b813::17"
    }
  },

  "vimNetwork": {

```

```

        "name": "ext-net",
        "ipVersion": "IPv4",
        "subnet": "ext-net-subnet",
        "fixedIps": {
            "vimIp": "10.75.189.199"
        }
    },
    "ntpServerIp": "10.250.32.10",
    "secondaryVnfmVolumeId": "eb66dc4d-ddf1-4880-875c-417d245f44d1",
    "vimRouteAddress": "10.75.167.0/24,10.75.185.0/24,10.75.171.192/26",
    "image": "VNFM_6.0.2.0.0_60.2.9",
    "flavor": "vnfm",
    "availabilityZone": "nova"
}
}

```

vnfConfigurableProperties Json:

FIXED IP Model (VNFM - V2)

```

{
    "ntpServerIp" : "10.250.32.10",
    "vimNetwork" : {
        "name" : "ext-net2",
        "ipVersion" : "IPV4",
        "subnet" : "ext-net2-subnet",
        "fixedIps":{
            "vimIp": "10.196.28.10"
        }
    },
    "secondaryVnfmVolumeId" : "7e8819f2-1f4f-4e6f-a02d-6e6763d8edc8",
    "vimRouteAddress" :
    "10.75.167.0/24,10.75.185.0/24,10.75.171.192/26,10.75.0.0/16,10.75.189.0/24,10.75.161.0/24"
}

```

Note

- The 202 response indicates that the request is accepted for processing. The VNF might take up to 15 minutes to become operational. Use the LCM operation ID to determine when the VNF is operational.
- The supported VNFM flavor is VNFM Secondary.
- The supported VNFM instantiation level ID is HA.
- Supported for IPv6 networks - ipVersion should be IPv6 in the request. The GUI can be accessed by the following URL: `https://[<VNFM SECONDARY IP>]:8443/docs/vnfm`.
Example: `https://[fd0d:deba:d97c:2c:6e41:6aff:fec7:80bf]:8443/docs/vnfm`.
- Primary and secondary VNFM creation can be done using the old VNFM volume.

10.6.2.1 Dynamic and Fixed IP Deployment Parameter Description

The following tables describes the parameters used for sending request to VNFM:

Table 10-5 Dynamic IP Deployment Parameters VNFM V1

Parameter	Description
flavourId	Identifies the VNFM Secondary deployment flavor to be instantiated.
vnfmNetwork	Provides network details to be used to access the VNFM (Swagger, ssh).
vimNetwork	Provides network details to be used to access VIM.
name	Provides the name of the respective network.
ipVersion	Identifies the IP version of the network, such as IPv4 or IPv6.
subnet	Identifies subnet of the respective network.
ntpServerIp	Identifies the IP of the NTP server.
secondaryVnfmVolumeId	Enters the volume name that can be used as a persistence storage for the VNFM.
vimRouteAddress	Enters the openstack network address or subnet mask. This is used for communication between VNFM and Openstack (Vim) network. Users can provide the list of route address separated by comma.
flavor (optional)	Provides the flavor used for openstack deployment.
image (optional)	Provides the image used for openstack deployment.
availabilityZone (optional)	Provides the name of logical partitioning in case of host aggregate.

Table 10-6 Parameters and Definitions

Parameter	Definitions
vnfConfigurableProperties	Data required for vnf configuration.
ntpServerIp	IP of the NTP server

Table 10-6 (Cont.) Parameters and Definitions

Parameter	Definitions
imiNetwork	Network used for internal communication of DSR entities.

Table 10-7 Fixed IP Deployment Parameters VNFM V1

Parameter	Description
flavourId	Identifies the VNFM Secondary deployment flavor to be instantiated.
vnfmNetwork	Provides network details to be used to access the VNFM (Swagger, ssh).
vimNetwork	Provides network details to be used to access VIM.
name	Provides the name of the respective network.
ipVersion	Identifies the IP version of the network, such as IPv4 or IPv6.
subnet	Identifies subnet of the respective network.
vnfmIp	Identifies the IP of the VNFM network interface.
vimIp	Identifies the IP of the VIM network interface.
ntpServerIp	Identifies the IP of the NTP server.
secondaryVnfmVolumeId	Enters the volume name that can be used as a persistence storage for the VNFM.
vimRouteAddress	Enters the openstack network address or subnet mask. This is used for communication between VNFM and Openstack (Vim) network. Users can provide the list of route address separated by comma.
flavor (optional)	Provides the flavor used for openstack deployment.
image (optional)	Provides the image used for openstack deployment.
availabilityZone (optional)	Provides the name of logical partitioning in case of host aggregate.

10.7 Instantiating the Network OAM VNF

Network OAM VNF supports both dynamic and fixed IP deployment.

To start a DSR deployment, it is required to instantiate a DSR network OAM VNF. Before deploying the VNF, make sure the following information is available:

The **VNF ID** for a previously created DSR Network OAM VNF instance.

Information about the OpenStack instance on which the VNF must be deployed:

- OpenStack Controller URI
- User Domain Name
- Project Domain Id
- Username
- Password
- Project name

The name of a Public Network in your chosen OpenStack instance that will carry the VNFM traffic..

The name of a network in the selected OpenStack instance that will carry the VIM traffic.

The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server, and is often a good choice.

For more information about the full list of all inputs and possible outputs of the **instantiate VNF** command, see **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the **DSR VNFM Swagger specification**.

Sample Request for DYNAMIC IP model (Dual Subnet))

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request>/instantiate`

Accept: `application/json`

Content-Type: `application/json`

Authorization: `<Token generated after login>`

Request JSON (VNFM - V1)

```
{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
    },
    "accessInfo": {
      "Project": "*****"
      "region": "regionone"
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default",
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "vipSubnetName": "ext-net-ipv6-subnet",

      "ipVersion": "IPv6",
      "name": "ext-net-ipv6-subnet"
    },
  },
}
```

```

        "ipVersion": "IPv4",
        "name": "ext-net-subnet"
    }],
    },
    "imiNetwork": {
        "name": "imi-net",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "test6"
        },
        {
            "ipVersion": "IPv4",
            "name": "test11"
        }
    ]
    },
    "ntpServerIp": "10.250.32.10",
    "flavor": "dsr.noam",
    "image": "DSR-9.0.2.0.0.vmdk",
    "availabilityZone": "nova",
    "noamAffinityPolicy": "anti-affinity"
}
}

```

vnfConfigurableProperties Json:

Request JSON (VNFM - V2)

```

{
  "ntpServerIp" : "10.250.32.10",
  "imiNetwork" : {
    "name" : "imiNetworkDual",
    "subnet" : [ {
      "name" : "test11",
      "ipVersion" : "IPv4"
    }, {
      "name" : "test6",
      "ipVersion" : "IPv6"
    } ]
  }
}

```

Sample Request for DYNAMIC IP model (Single Subnet)

Note

- vnfm_version is either v1 or v2.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login

Request JSON (VNFM - V1)

```
{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [],

  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default",
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
    },
    "subnet": [
      {
        "ipVersion": "IPv4",
        "name": "ext-net-subnet"
      }
    ],
    "imiNetwork": {
      "name": "imi-net",
      "subnet": [
        {
          "ipVersion": "IPv4",
          "name": "test11"
        }
      ]
    },
    "ntpServerIp": "10.250.32.10",
    "flavor": "dsr.noam",
    "image": "DSR-9.0.2.0.0.vmdk",
    "availabilityZone": "nova",
    "noamAffinityPolicy": "anti-affinity"
  }
}
```

vnfConfigurableProperties**Request JSON (VNFM - V2)**

```
{  "ntpServerIp" : "10.250.32.10",
    "imiNetwork" :
    {  "name" : "imiNetworkDual",
        "subnet" : [ {  "name" : "test11",
                        "ipVersion" : "IPv4"
                      }
                ]
    }
}
```

Sample Request for FIXED IP deployment model

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated login>

Request JSON (VNFM - V1)

```
{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],

  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default",
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "subnet": [{
        "ipVersion": "IPv4",
        "name": "ext-net-subnet",

        "fixedIps": {  "primaryNoamIp": "10.75.189.224",
```

```

        "secondaryNoamIp": "10.75.189.236",
        "noamVip": "10.75.189.238"
    },
    "ipVersion": "IPv6",
    "name": "ext-net-ipv6-subnet",
    "fixedIps": {
        "primaryNoamIp": "2606:b400:605:b818:6e41:6aff:fec7:80e0",
        "secondaryNoamIp": "2606:b400:605:b818:6e41:6aff:fec7:80f9"
    }
}
]
},

    },
    "imiNetwork": {
        "name": "imi-net",
        "subnet": [{
            "ipVersion": "IPv4",
            "name": "ext-net-subnet",
            "fixedIps": {
                "primaryNoamImiIp": "10.75.189.224",
                "secondaryNoamImiIp": "10.75.189.236"
            }
        }
    ],
    {
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet",
        "fixedIps": {
            "primaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80e0",
            "secondaryNoamImiIp": "2606:b400:605:b818:6e41:6aff:fec7:
80f9"
        }
    }
]
},
    "ntpServerIp": "10.250.32.10",
    "flavor": "dsr.noam",
    "image": "DSR-9.0.2.0.0.vmdk",
    "availabilityZone": "nova",
    "noamAffinityPolicy": "anti-affinity"
}
}

```

vnfConfigurableProperties Json:**Request JSON (VNFM - V2)**

```

{
  "ntpServerIp" : "10.250.32.10",
  "imiNetwork" : {
    "name" : "imiNetworkDual",
    "subnet" : [ {
      "name" : "test11",
      "ipVersion" : "IPv4",
      "fixedIps" : {

```

```

        "primaryNoamImiIp" : "192.168.0.66",
        "secondaryNoamImiIp" : "192.168.0.252"
      }, {
        "name" : "test6",
        "ipVersion" : "IPv6",
        "fixedIps" : {
          "primaryNoamImiIp" : "2001:db8:1231::fc",
          "secondaryNoamImiIp" : "2001:db8:1231::234"
        }
      }
    ]
  }
}

```

Note

: User's responsibility to identify available IP addresses in the network to be used. If the user provides a IP address which is not available the subnet, stack creation fails.

Sample Response: Instantiating NOAM Request.

```

202 Accepted
Headers:
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0  content-type:
  application/xml
}

```

Note

- The 202 response means that the request was accepted for processing. The VNF might take up to 15 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.
- If the VNFM creates a VNF that is operational, but has no Signaling VNFs, then it is required to deploy one or more Signaling VNF, and create the DIAMETER configuration data (peers, connections, etc.) for those VNFs, to perform DIAMETER routing.
- After NOAM VNF deployment, the standby NOAM is automatically changed to **Force StandBy**, purposely to avoid any switchover, while DSR Signaling VNF is deployed. Once DSR Signaling Site is deployed and no more Life Cycle Management operations are planned, change **Force Standby** NOAM to Active by changing the **Max Allowed HA Role** to **Active** on the **Status & Manage** -> **HA** options in the Active NOAM GUI.
- The supported NOAM Flavor is **DSR NOAM**.
- The supported NOAM instantiation level id is **HA**, that creates two NOAMs.
- Supported for IPv6 networks - ipVersion should be "IPv6" in the request Body. The GUI can be accessed by the following URL: [https://\[<NOAM-vip>\]](https://[<NOAM-vip>]) .
For example: [https://\[fd0d:deba:d97c:2c:6e41:6aff:fec7:80bf\]](https://[fd0d:deba:d97c:2c:6e41:6aff:fec7:80bf])

Expected Alarms:

10073 Server Group Max Allowed HA Role Warning

Resolution: This alarm can be resolved by, **Status and Manage Server tab** → **HA** → **changing Max HA Role** field of StandBy NOAM to active.

The following table describes the parameters used for sending request to VNFM.

Table 10-8 Parameters and Definitions for Network OAM VNF VNFM V1

Parameters	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used for internal communication of DSR entities
ipVersion	IP version of the network - "IPv4"/"IPv6"
ntpServerIp	IP of the NTP server
flavor (optional)	flavor used for openstack deploys
image (optional)	image used for openstack deploys
vipSubnetName (In case of Dual Subnet)	Name of VIP subnet to be used only in case of Dual Subnet
availabilityZone (optional)	name of logical partitioning in case of host aggregate
noamAffinityPolicy (optional)	openstack affinity policy for NOAM

Parameters and Definitions for Dynamic and Fixed IP deployment (VNFM-V2)**Table 10-9 Parameters and Definitions**

Parameter	Definitions
vnfConfigurableProperties	Data required for vnf configuration.
ntpServerIp	IP of the NTP server
imiNetwork	Network used for internal communication of DSR entities.

Parameters and Definitions for FIXED IP deployment (VNFM - V1)**Table 10-10 Parameter and Definitions**

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavour to be instantiated
xmiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used for internal communication of DSR entities
ipVersion	IP version of the network - "IPv4"/"IPv6"
ntpServerIp	IP of the NTP server
fixedIps	Json object in network to provide IP address
primaryNoamIp	IP address for primary NOAM IP
secondaryNoamIp	IP address for secondary NOAM IP
noamVip	IP address for NOAM VIP
primaryNoamImip	IP address for primary NOAM IP of IMI
secondaryNoamImip	IP address for secondary NOAM IP of IMI
flavor (optional)	flavor used for openstack deploys
image (optional)	image used for openstack deploys
availabilityZone (optional)	name of logical partitioning in case of host aggregate
noamAffinityPolicy (optional)	openstack affinity policy for NOAM

10.8 Instantiating the DR Network OAM VNF (VNFM-V1)

DR NOAM is the Disaster recovery NOAM site. The operator can make DRNOAM as the Primary Site, in case both the Active and StandBy NOAM of Primary site fails, and can continue the operations without any disturbance.

DRNOAM supports both dynamic and fixed deployment model.

When a setup is configured with a DR NOAM then first NOAM SG is treated as Primary NOAM Site and second NOAM SG is treated as Secondary NOAM site.

To instantiate a DSR DR Network OAM VNF, the following information must be available:

- The **VNF ID** for a previously created DSR DR Network OAM VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:

- OpenStack endpoint URI
- User Domain Name
- Project Domain Id
- Username
- Password
- Project name
- The name of a Public Network in your chosen OpenStack instance that will carry OAM traffic.
- OpenStack resource IDs for the XMI IPs from both DSR NOAM VMs.

Note

The resource IDs can be obtained by examining the DSR Network OAM stack to which the identified DR NOAM VNF would be attached.

- Name of Active Primary DSR NOAM VM.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server, and is often a good choice.

10.8.1 Determining the DR NOAM XMI Resource IDs

The following facts must be considered before proceeding with DR NOAM site creation:

- DRNOAM site must be created on separate tenant.
- DRNOAM site is referred as Secondary NOAM. Therefore, we have two sites, Primary and Secondary.
- Secondary Site configuration is done on Primary Active NOAM.
- In the Primary Active NOAM, when second NOAM Server Group gets created, it automatically becomes Secondary.
- Primary Active NOAM communicates to Secondary Active NOAM through the existing comcol replication and merging mechanism.
- Secondary NOAM Site is optional and it does not need to be deployed at the same time as of Primary NOAM.

From the OpenStack GUI, to change your view to the tenant on which the DSR Network OAM VNF is deployed, perform the following steps.

1. Go to **Project->Network->Network Topology**. A diagram of all VMs in the tenant is displayed.

Note

The diagram may take few minutes to display.

2. Click one of the NOAM VMs. A pop-up appears having information about the specific NOAM VM.
3. Save the resource ID for the XMI port provided in the IP Addresses section of the pop-up.

Note

The IP Addresses section of the popup contains information about the network ports and resource IDs, assigned to the VM.

4. Repeat the previous step for the other NOAM VM.

You can also use the following alternative:

- Instead of passing resource IDs, user can use DSR-NOAM XMI IPs.
- User can pass Active DSR-NOAM's XMI IP to resource id 1 and StandBy DSR-NOAM's XMI IP to resource id 2.

For more information about the full list of all inputs and possible outputs of the **instantiate VNF** command, see **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the **DSR VNFM Swagger specification**.

Swagger specifications can be found post VNFM installation at (https://<VNFM_IP>:8443/docs/vnfm/).

Sample Request

Resource URL: https://<myhost-IP>:8443/vnflcm/<vnfm_version>/vnf_instances/<VNF_ID_received_from_create_request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization: Token generated after login

Instantiating DR NOAM Request for Dynamic IP deployment.

Request JSON (VNFM - V1)

```
{
  "flavourId": "DSR DR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "id1",
    "virtualLinkDescId": " Active NOAM",
    "resourceId": "156d73cf-6e44-456b-a661-14bd0cc2b43c"
  },
  {
    "id": "id2",
    "virtualLinkDescId": " StandBy NOAM",
    "resourceId": "5c638770-5585-44c7-97c7-b4a52a26e5ec"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
```

```

        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default",
    }
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net3",
        "vipSubnetName": "ext-net-ipv6-subnet",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "ext-net-ipv6-subnet"
        },
        {
            "ipVersion": "IPv4",
            "name": "ext-net-subnet"
        }
    ]
},
    "imiNetwork": {
        "name": "imi-net",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "test6"
        },
        {
            "ipVersion": "IPv4",
            "name": "test11"
        }
    ]
},
    "ntpServerIp": "10.250.32.10",
    "primaryNoamVmName": "NOAM00-ea47f4b1",
    "flavor": "dr.noam",
    "image": "DSR-9.0.2.0.0.vmdk",
    "availabilityZone": "nova",
    "drNoamAffinityPolicy": "anti-affinity"
}
}
}
}]]

```

Note

The "vipSubnetName" field is used only in case of Dual Subnet.

vnfConfigurableProperties Json VNFM V2:

```

{
    "ntpServerIp": "10.250.32.10",
    "imiNetwork": {
        "name": "imiNetworkDual",

```

```

        "subnet": [{
            "name": "test11",
            "ipVersion": "IPv4"
        }, {
            "name": "test6",
            "ipVersion": "IPv6"
        }]
    },
    "primaryNoamVmName": "FT--453a634e-DN000",
    "noamXmiIps": ["10.75.234.65", "10.75.234.99"]
}

```

Instantiating DR NOAM Request for Fixed IP deployment.

```

{
    "flavourId": "DSR DR NOAM",
    "instantiationLevelId": "HA",
    "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [
        {
            "id": "id1",
            "virtualLinkDescId": "Active NOAM IP's",
            "resourceId": "38121fc6-310c-4012-9787-b5289dd620b9"
        },
        {
            "id": "id2",
            "virtualLinkDescId": "Secondary NOAM IP's",
            "resourceId": "baa54c8d-1a7a-4b15-8d64-8fe9af50b000"
        }
    ],
    "vimConnectionInfo": {
        "MvlCloud": {
            "vimid": "vimid",
            "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
            "interfaceInfo": {
                "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
            },
            "accessInfo": {
                "project": "*****",
                "region": "RegionOne",
                "username": "*****",
                "password": "*****",
                "userDomain": "Default",
                "projectDomain": "default",
            }
        }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
        "ntpServerIp": "10.250.32.10",
        "xmiNetwork": {
            "name": "ext-net4",
            "vipSubnetName": "ext-net-ipv6-subnet",
            "subnet": [{

```

```

        "ipVersion": "IPv4",
        "name": "ext-net-subnet",
        "fixedIps": {
            "drPrimaryNoamIp": "10.75.189.224",
            "drSecondaryNoamIp": "10.75.189.236",
            "drNoamVip": "10.75.189.238"
        }
    },
    {
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet",
        "fixedIps": {
            "drPrimaryNoamIp":
"2606:b400:605:b818:6e41:6aff:fec7:80e0",
            "drSecondaryNoamIp":
"2606:b400:605:b818:6e41:6aff:fec7:80f9"
        }
    }
}],
    "imiNetwork": {
        "name": "imi-net",

        "subnet": [{
            "ipVersion": "IPv4",
            "name": "ext-net-subnet",
            "fixedIps": {
                "drPrimaryNoamImiIp": "10.75.189.224",
                "drSecondaryNoamImiIp": "10.75.189.236"
            }
        },
        {
            "ipVersion": "IPv6",
            "name": "ext-net-ipv6-subnet",
            "fixedIps": {
                "drPrimaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80e0",
                "drSecondaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80f9"
            }
        }
    ],
    "primaryNoamVmName": "NOAM00-9ca5c163",

    "flavor": "dr.noam",
    "image": "DSR-9.0.2.0.0.vmdk",
    "availabilityZone": "nova",
    "drNoamAffinityPolicy": "anti-affinity"
}
}

```

vnfConfigurableProperties Json:

Request JSON (VNFM - V2)

```
{
  "ntpServerIp": "10.250.32.10",
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "name": "test11",
      "ipVersion": "IPv4",
      "fixedIps": {
        "drPrimaryNoamImiIp": "10.75.189.224",
        "drSecondaryNoamImiIp": "10.75.189.236"
      }
    }, {
      "name": "test6",
      "ipVersion": "IPv6",
      "fixedIps": {
        "drPrimaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80e0",
        "drSecondaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80f9"
      }
    }
  ]
},
  "primaryNoamVmName": "FT--453a634e-DN000",
  "noamXmiIps": ["10.75.234.65", "10.75.234.99"]
}
```

Sample Response: Instantiating DR NOAM Response.

202 Accepted

Headers:

```
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 21 Feb 2019 10:39:24 GMT
  content-length: 0  content-type:
  application/xml
}
```

Note

- The 202 response means that the request was accepted for processing. The VNF might take up to 15 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.
- The supported NOAM Flavor is **DSR NOAM Flavor**.
- The supported DSR DR NOAM instantiation level id is **HA**.
- Support for IPv6 networks - ipVersion should be "IPv6" in the request Body.

Table below describes the parameters used for sending request to VNFM.

Table 10-11 Parameters and Definitions for dynamic IP deployment (VNFM - V1)

Parameters	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated
instantiationLevelId	Identifier of the instantiation level of the deployment flavor to be instantiated. If not present, the default instantiation level is HA.
resourceId	The identifier of the resource in the scope of the VIM or the resource provider
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used for internal communication of DSR entities
name	Network name, for example; ext-net
ipVersion	IP version IPv4 or IPv6
ntpServerIp	IP of the NTP server
primaryNoamVmName	Primary Active DSR NOAM VM name
flavor (optional)	flavor used for openstack deploys
image (optional)	image used for openstack deploys
availabilityZone (optional)	name of logical partitioning in case of host aggregate
vipSubnetName (In case of Dual Subnet)	Name of VIP subnet to be used only in case of Dual Subnet
drNoamAffinityPolicy (optional)	Openstack affinity policy for DR NOAM

Parameters and Definitions for dynamic and Fixed IP deployment (VNFM - V2)**Table 10-12 Parameters and Definitions**

Parameter	Definitions
vnfConfigurableProperties	Data required for vnf configuration.
ntpServerIp	IP of the NTP server.
imiNetwork	Network used for internal communication of DSR entities.
primaryNoamVmname	Primary Active DSR NOAM VM name.
noamXmilps	Primary and Secondary Noam XMI IP addresses.

10.9 Multiple XSI Support (1 - 16 XSI interfaces) - VNFM V1

VNFM V1 supports multiple xsi from 1 to 16 , there is no restriction from VNFM side to pass any specific number of xsi's while creating DSR Signaling setups.

This feature helps in getting rid of VNFM v1 limitation of using only 1, 2 and 4 XSI.

It supports both fixed and dynamic IP deployment model for all flavors of Signaling VNFs.

Sample Request for signaling flavor Diameter with multiple xsi (5 xsi interface) for DYNAMIC IP (Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/ < VNF ID
received from create request > /instantiate

Accept: application/json

Content Type: application/json

Request Json

Language: json

theme: confluence

firstline: 1

title: Signaling DIAMETER with multiple xsi (5 xsi interfaces) dynamic
IP(Dual Subnet) (VNFM - V1)

linenumbers true

collapse: true

```
{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "vipSubnetName": "ext-net-ipv6-subnet",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
      }
    ]
  }
}
```

```

        },
        {
            "ipVersion": "IPv4",
            "name": "ext-net-ipv4-subnet"
        }
    ]
},
"imiNetwork": {
    "name": "imi-private",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6"
    }],

    {
        "ipVersion": "IPv4",
        "name": "test4"
    }
]
},
"xsiNetwork": [{
    "name": "ext-net2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiIPv6"
    }],

    {
        "ipVersion": "IPv4",
        "name": "xsiIPv4"
    }
]
},
{
    "name": "xsiNetworkDual2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual2-IPv6"
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual2-IPv4"
    }
]
},
{
    "name": "xsiNetworkDual3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6"
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual3-IPv4"
    }
]
}

```

```

    }
  ],
  {
    "name": "ext-net5-dual",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext-net5-IPv6"
    },
    {
      "ipVersion": "IPv4",
      "name": "ext-net5-IPv4"
    }
  ]
},
{
  "name": "ext-net7",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "ext-net7-IPv6"
  },
  {
    "ipVersion": "IPv4",
    "name": "ext-net7-IPv4"
  }
]
}
],
"ntpServerIp": "10.250.32.10",
"primaryNoamVmName": "NOAM00-32cd6138",
"noamSgName": "dsrNetwork0am_NOAM_32cd6138_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
"soamAvailabilityZone": "nova",
"ipfeFlavor": "dsr.ipfe",
"ipfeImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
"ipfeAvailabilityZone": "nova",
"daFlavor": "dsr.da",
"daImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
"daAvailabilityZone": "nova",
"stpFlavor": "dsr.stp",
"stpImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
"stpAvailabilityZone": "nova",
"soamAffinityPolicy": "anti-affinity",
"ipfeAffinityPolicy": "anti-affinity",
"daAffinityPolicy": "soft-anti-affinity"
}
}

```

Note

vipSubnetName field to be used only in case of Dual Subnet.

10.10 Instantiating the Signaling VNF with Multiple XSI (1, 2, and 4 XSI Interface)

Signaling VNF supports both dynamic and fixed IP deployment

To deploy the first signaling VNF, the following must be available:

A previously instantiated DSR Network OAM VNF.

The VNF ID for a previously created DSR Signaling VNF instance.

Information about the OpenStack instance on which you want to deploy the VNF:

- OpenStack endpointURI
- User Domain Name
- Project Domain Id
- Username
- Password
- Project name

The name of a Public Network in your chosen OpenStack instance that will carry OAM traffic.

The name of a Public Network in your chosen OpenStack instance that will carry Signaling traffic.

Note

This should be a different network than the one that carries OAM traffic.

The IP address of the NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls your chosen OpenStack instance normally hosts an NTP server, and is often a good choice.

OpenStack resource IDs for the XMI IPs from both NOAM VMs.

Note

The resource IDs can be obtained by examining the network OAM stack to which the identified signaling VNF would be attached .

Name of the active NOAM VM.

Note

To avoid switchover of Active NOAM, make the StandBy NOAM as **Forced Standby** by changing the **Max Allowed HA Role** to **Standby** on **Status & Manage -> HA** from Active NOAM GUI.

Name of the NOAM SG.

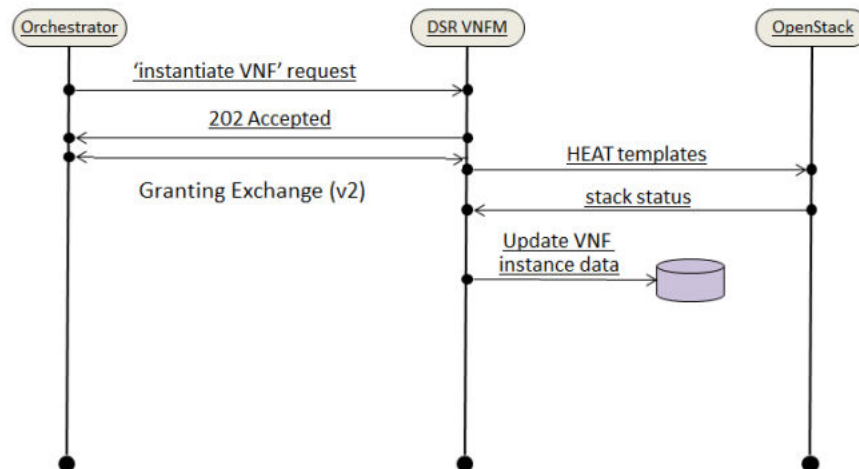
Expected Alarms:

IPFE Alarm: 5002 An address pertaining to inter-IPFE state synchronization is configured incorrectly.

DAMP Alarm: 19800 Communication Agent Connection Down (occurs only in the presence of SBR servers).

The following image illustrates the VNF instantiation:

Figure 10-5 VNF installation



The following table contains the supported Instantiation levels to instantiate a VNF resource for the DSR Signaling VNF.

Table 10-13 Supported Instantiation Levels for DSR Signaling VNF

VNFM Signaling Flavors	Small						Medium						Large					
	DA MP	IPF E	SE RV IC E MP	ST P	SB R	UD R	DA MP	IPF E	SE RV IC E MP	ST P	SB R	UD R	DA MP	IPF E	SE RV IC E MP	ST P	SB R	UD R
DIAMETER	2	2	0	0	0	0	4	2	0	0	0	0	32	4	0	0	0	0
STP	0	0	2	2	0	0	0	0	4	8	0	0	0	0	8	32	0	0
DIAMETER+STP	2	2	2	2	0	0	4	2	4	8	0	0	24	4	8	8	0	0
DIAMETER+SBR	2	2	0	0	3	0	4	2	0	0	6	0	32	4	0	0	9	0
DIAMETER+UDR	2	2	0	0	0	2	4	2	0	0	0	2	32	4	0	0	0	4

Table 10-13 (Cont.) Supported Instantiation Levels for DSR Signaling VNF

VNFM Signaling Flavors	Small						Medium						Large					
	DAMP	IPFE	SE RV IC E MP	STP	SBR	UDR	DAMP	IPFE	SE RV IC E MP	STP	SBR	UDR	DAMP	IPFE	SE RV IC E MP	STP	SBR	UDR
STP+UDR	0	0	2	2	0	2	0	0	4	8	0	2	0	0	8	32	0	4

Note

- DSR Signaling supports "custom" instantiation level ID based on flavor.
- In case of UDR flavors, VNFM supports one and two xsi interfaces for UDR.
- Total number of servers allowed per signaling VNF is 48.
- Total number of IPFE servers allowed per signaling VNF is 4.
- Total number of SOAMs for any of the above servers is 2.

For Example: Total number of servers per signaling VNF = No. of SOAM's + No. of DAMP's + No. of IPFE's + No. of STP's + No. of SBR's+ No. of UDR's.

For SBR flavors:

- In case of SBR flavors, it is mandatory to pass the `sbrNetwork` parameter for instantiation of signaling stack. VNFM always creates Replication port for SBRs.
- Breakdown on the number of SBRs :
 1. SMALL - 3 SBRs, 2 SBRs will be added to one server group and one is not added. Server group will have 2 SBRs.
One SBR will be left out to be added manually in the mated site's server group.
 2. MEDIUM - 6 SBRs, 2 Server Groups.
Each server group will have 2 SBRs.
One SBR from each server group will be left out to be added manually in the mated site's server groups.
 3. LARGE - 9 SBRs, 3 Server Groups.
Each server group will have 2 SBRs.
One SBR from each server group will be left out to be added manually in the mated site's server groups.
- The SBRs created are plain, they have to be manually configured as Session, Binding or Universal.

Note

UDR Flavors

- In case of UDR flavors, VNFM currently supports one and two xsi interfaces for UDR

Note

STP Flavors (Service Mp)

- Service Mp feature is only supported for STP flavors.

10.10.1 DSR Signaling VNF with Multiple XSI Support (1, 2 and 4 xsi interface only)

- Multiple XSI supports only DSR Signaling VNF.
- DAMP vnf supports 1, 2 and 4 xsi interface.
- STPMP vnf supports 1, 2, and 4 xsi interface.
- IPFE vnf supports 1, 2, and 4 xsi interface.
- UDR vnf supports only 1 and 2 xsi interface.

While passing xsiNetwork through the request body, add the list of network in xsiNetwork.

For Example

1 xsiNetwork	2 xsiNetwork	4 xsiNetwork
<pre>"xsiNetwork": [{ "name": "provider- vlan500", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }]</pre>	<pre>"xsiNetwork": [{ "name": "provider- vlan500", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }, { "name": "provider- vlan610", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }]</pre>	<pre>"xsiNetwork": [{ "name": "provider- vlan500", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }, { "name": "provider- vlan610", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }, { "name": "provider- vlan500", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }, { "name": "provider- vlan610", "subnet": [{ "name": "<subnet-name>", "ipVersion": "IPv4" }] }]</pre>

Note

In case of the UDR flavor (Diameter+UDR, STP+UDR), "udrType" is an optional parameter. So, if a user has not used it in the request message, by default, UDR of type "EIR" is created. The other UDR type is "vMNP".

The sample request and response provided below represents signaling flavors without SBR such as, DIAMETER, STP and DIAMETER+STP, DIAMETER+UDR, and STP+UDR, with multiple xsi (1, 2, 4 xsi interface) for Dynamic IP and Fixed IP deployment model.

Sample Request

Resource URL: URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Instantiating the first signaling VNF request for Dynamic IP (Dual Subnet) deployment model (VNFM-V1)

```
{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "vipSubnetName": "ext-net-ipv6-subnet",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
      }],
    },
  },
}
```

```

        {
            "ipVersion": "IPv4",
            "name": "ext-net-ipv4-subnet"
        }
    ]
},
"imiNetwork": {
    "name": "imi-private",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6"
    }],

    {
        "ipVersion": "IPv4",
        "name": "test4"
    }
]
},
"xsiNetwork": [{
    "name": "ext-net2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiIPv6"
    }],

    {
        "ipVersion": "IPv4",
        "name": "xsiIPv4"
    }
]
},
{
    "name": "xsiNetworkDual2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual2-IPv6"
    }],
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual2-IPv4"
    }
]
}
],
"ntpServerIp": "10.250.32.10",
"primaryNoamVmName": "NOAM00-32cd6138",
"noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0.vmdk",
"soamAvailabilityZone": "nova",
"ipfeFlavor": "dsr.ipfe",
"ipfeImage": "DSR-9.0.2.0.0.vmdk",

```

```

        "ipfeAvailabilityZone": "nova",
        "daFlavor": "dsr.da",
        "daImage": "DSR-9.0.2.0.0.vmdk",
        "daAvailabilityZone": "nova",
        "stpFlavor": "dsr.stp",
        "stpImage": "DSR-9.0.2.0.0.vmdk",
        "stpAvailabilityZone": "nova",
        "soamAffinityPolicy": "anti-affinity",
        "ipfeAffinityPolicy": "anti-affinity",
        "daAffinityPolicy": "soft-anti-affinity"
    }
}

```

Note

The "vipSubnetName" field is used only in case of Dual Subnet.

vnfConfigurableProperties (VNFM-v2):

Signaling DIAMETER with multiple xsi(4 xsi interface) for dynamic IP(Dual Subnet) (VNFM - V2)

```

{
    "ntpServerIp": "10.250.32.10",
    "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [{
            "name": "test11",
            "ipVersion": "IPv4"
        },
        {
            "name": "test6",
            "ipVersion": "IPv6"
        }
    ]
    },
    "xsiNetworkCount": 4,
    "primaryNoamVmName": "csarNo-DN000",
    "noamSgName": "csarNo_DN0SG",
    "noamXmiIps": [
        "10.75.189.150",
        "10.75.189.212"
    ]
}

```

Sample Request for signaling flavor DIAMETER with multiple xsi(4 xsi interface) for FIXED IP(Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling flavor DIAMETER with multiple xsi(4 xsi interface) for FIXED IP(Dual Subnet) (VNFM - V2)

```
{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 4,
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "name": "test11",
      "ipVersion": "IPv4",
      "fixedIps": {
        "primarySoamImiIp": "10.75.189.170",
        "secondarySoamImiIp": "10.75.189.171",
        "dampImiIps": [
          "10.75.189.172",
          "10.75.189.173"
        ],
        "ipfeImiIps": [
          "10.75.189.174",
          "10.75.189.175"
        ]
      }
    }
  },
  {
    "name": "test6",
    "ipVersion": "IPv6",
    "fixedIps": {
      "primarySoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:77d0",
      "secondarySoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:76d0",
      "dampImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:75d0",
        "2606:b400:605:b818:6e41:6aff:fec7:74d0"
      ],
      "ipfeImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:73d0",
        "2606:b400:605:b818:6e41:6aff:fec7:72d0"
      ]
    }
  }
]
```

Sample Request for signaling flavor DIAMETER with multiple xsi(2 xsi interface) for DYNAMIC IP(Single Subnet - Ipv6) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling DIAMETER with multiple xsi(2 xsi interface) for DYNAMIC IP(Single Subnet - Ipv6) (VNFM - V2)

```
{
  "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
  "xsiNetworkCount": 2,
  "imiNetwork": {
    "name": "imi6-private",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "imi6-sub"
    }]
  },
  "primaryNoamVmName": "csarNo-DN000",
  "noamSgName": "csarNo_DN0SG",
  "noamXmiIps": ["10.75.189.150", "10.75.189.212"]
}
```

Sample Request for signaling flavor DIAMETER with multiple xsi(2 xsi interface) for FIXED IP(Single Subnet - Ipv6) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling DIAMETER with multiple xsi(2 xsi interface) for FIXED IP(Single Subnet - Ipv6) (VNFM - V2) {

```
"ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
"imiNetwork": {
  "name": "imi6-private",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "imi6-sub",
    "fixedIps": {
      "primarySoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:77d0",
      "secondarySoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:76d0",
      "dampImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:75d0",
        "2606:b400:605:b818:6e41:6aff:fec7:74d0"
      ],
      "ipfeImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:73d0",
        "2606:b400:605:b818:6e41:6aff:fec7:72d0"
      ]
    }
  ]
},
},
```

```

    "xsiNetworkCount": 2,
    "primaryNoamVmName": "csarNo-DN000",
    "noamSgName": "csarNo_DNOSG",
    "noamXmiIps": ["10.75.189.150", "10.75.189.212"]
  }

```

Sample Request for signaling flavor DIAMETER with "custom" instantiationLevelId, 1 xsi DYNAMIC IP (Single ipv4 Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling flavor DIAMETER with "custom" instantiationLevelId, 1 xsi DYNAMIC IP (Single ipv4 Subnet) (VNFM - V2)

```

{
  "ntpServerIp": "10.250.32.10",
  "primaryNoamVmName": "vimNO-DN000",
  "noamSgName": "vimNO_DNOSG",
  "xsiNetworkCount": 1,
  "noamXmiIps": [
    "10.75.189.231",
    "10.75.189.211"
  ],
  "numDa": "1",
  "numIpfe": "1",
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "ipVersion": "IPv4",
      "name": "test11"
    }]
  }
}

```

Sample Request for signaling flavor DIAMETER with "custom" instantiationLevelId, 1 xsi FIXED IP (Single ipv4 Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling flavor DIAMETER with "custom" instantiationLevelId, 1 xsi FIXED IP (Single ipv4 Subnet) (VNFM - V2)

```

{
  "ntpServerIp": "10.250.32.10",
  "primaryNoamVmName": "vimNO-DN000",

```

```

    "noamSgName": "vimNO_DNOSG",
    "xsiNetworkCount": 1,
    "noamXmiIps": [
        "10.75.189.231",
        "10.75.189.211"
    ],
    "numDa": "1",
    "numIpfe": "1",
    "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [{
            "ipVersion": "IPv4",
            "name": "test11",
            "fixedIps": {
                "primarySoamImiIp": "10.75.189.181",
                "secondarySoamImiIp": "10.75.189.182",
                "dampImiIps": [
                    "10.75.189.185",
                    "10.75.189.186"
                ],
                "ipfeImiIps": [
                    "10.75.189.187",
                    "10.75.189.188"
                ]
            }
        ]
    }
}

```

Sample Request for signaling flavor DIAMETER in Mixed Mode with multiple xsi(4 xsi interface) for DYNAMIC IP (Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling DIAMETER in Mixed Mode with multiple xsi(4 xsi interface) for dynamic IP(Dual Subnet) (VNFM - V2)

```

{
    "ntpServerIp": "10.250.32.10",
    "xsiNetworkCount": 4,
    "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [{
            "name": "test11",
            "ipVersion": "IPv4"
        }
    ]
},
    "primaryNoamVmName": "csarNo-DN000",
    "noamSgName": "csarNo_DNOSG",
}

```

```

        "noamXmiIps": [
            "10.75.189.150",
            "10.75.189.212"
        ]
    }

```

Sample Request for signaling flavor DIAMETER in Mixed Mode with multiple xsi(4 xsi interface) for FIXED IP(Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling flavor DIAMETER in Mixed Mode with multiple xsi(4 xsi interface) for FIXED IP(Dual Subnet) (VNFM - V2)

```

{
    "ntpServerIp": "10.250.32.10",
    "xsiNetworkCount": 4,
    "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [{
            "name": "test11",
            "ipVersion": "IPv4",
            "fixedIps": {
                "primarySoamImiIp": "10.75.189.170",
                "secondarySoamImiIp": "10.75.189.171",
                "dampImiIps": [
                    "10.75.189.172",
                    "10.75.189.173"
                ],
                "ipfeImiIps": [
                    "10.75.189.174",
                    "10.75.189.175"
                ]
            },
            "ipVersion": "IPv4",
            "fixedIps": {
                "primarySoamImiIp": "10.75.189.170",
                "secondarySoamImiIp": "10.75.189.171",
                "dampImiIps": [
                    "10.75.189.172",
                    "10.75.189.173"
                ],
                "ipfeImiIps": [
                    "10.75.189.174",
                    "10.75.189.175"
                ]
            }
        ]
    },
    "primaryNoamVmName": "csarNo-DN000",
    "noamSgName": "csarNo_DN0SG",
    "noamXmiIps": [
        "10.75.189.150",
        "10.75.189.212"
    ]
}

```

vSTP Service MP deployment

- VNFM deploys the Service MP along with vSTP-MP only.
- Service MP deployment is optional. vSTP-MP can be independently deployed without service MP.

- In Signaling request body we need to pass "serviceMp": "true" . Default value is "false".
- If Service Mp was not included ("serviceMp": "false") while vSTP-MP instantiation, we can scale later Service Mp from Sizing 0 to any number (till maximum supported size). This we can only do by scale using "scaleInfo (Arbitrary Size)" operation.

Note

Service Mp Instantiation and scale introduced from DSR version **"8.5.0.1.0-91.14.0"**

Sample Request for signaling flavor DIAMETER + STP without serviceMp, multiple xsi(2 xsi interface) for DYNAMIC IP(Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling flavor DIAMETER + STP without serviceMp, multiple xsi(2 xsi interface) for DYNAMIC IP(Dual Subnet)(VNFM - V2)

```
{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 2,
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "name": "test11",
      "ipVersion": "IPv4"
    }, {
      "name": "test6",
      "ipVersion": "IPv6"
    }]
  },
  "primaryNoamVmName": "csarNoam-DN000",
  "noamSgName": "csarNoam_DN0SG",
  "noamXmiIps": ["10.75.189.249", "10.75.189.218"]
}
```

Sample Request for signaling flavor DIAMETER + STP without serviceMp, multiple xsi(2 xsi interface) for FIXED IP (Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling flavor DIAMETER + STP without serviceMp, multiple xsi(2 xsi interface) for FIXED IP(Dual Subnet) (VNFM - V2)

```

{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 2,
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "name": "test11",
      "ipVersion": "IPv4",
      "fixedIps": {
        "primarySoamImiIp": "10.75.189.161",
        "secondarySoamImiIp": "10.75.189.162",
        "dampImiIps": [
          "10.75.189.163",
          "10.75.189.164"
        ],
        "ipfeImiIps": [
          "10.75.189.165",
          "10.75.189.166"
        ],
        "stpImiIps": [
          "10.75.189.167",
          "10.75.189.168"
        ]
      }
    ]
  }, {
    "name": "test6",
    "ipVersion": "IPv6",
    "fixedIps": {
      "primarySoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80d0",
      "secondarySoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:79d0",
      "dampImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:78d0",
        "2606:b400:605:b818:6e41:6aff:fec7:81d0"
      ],
      "ipfeImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:77d0",
        "2606:b400:605:b818:6e41:6aff:fec7:81d0"
      ],
      "stpImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:76d0",
        "2606:b400:605:b818:6e41:6aff:fec7:75d0"
      ]
    }
  ]
}, {
  "primaryNoamVmName": "csarNoam-DN000",
  "noamSgName": "csarNoam_DN0SG",
  "noamXmiIps": ["10.75.189.249", "10.75.189.218"]
}

```

Sample Request (for signaling flavors without SBR i.e., DIAMETER,STP and DIAMETER+STP, DIAMETER+UDR, STP+UDR) with multiple xsi (1,2,4 xsi interface) for dynamic IP deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json Authorization :

Authorization : <Token generated after login>

Request JSON (VNFM - V1)

```
{
  "flavourId": "DIAMETER+STP",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
      }]
    },
    "imiNetwork": {
      "name": "imi-private",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6"
      }]
    }
  },
}
```

```

    "xsiNetwork": [{
      "name": "ext-net2",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiIPv6"
      }]
    },
    {
      "name": "xsiNetworkDual2",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual2-IPv6"
      }]
    }
  ],
  "ntpServerIp": "10.250.32.10",
  "primaryNoamVmName": "NOAM00-32cd6138",
  "noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
  "soamFlavor": "dsr.soam",
  "soamImage": "DSR-9.0.2.0.0.vmdk",
  "soamAvailabilityZone": "nova",
  "ipfeFlavor": "dsr.ipfe",
  "ipfeImage": "DSR-9.0.2.0.0.vmdk",
  "ipfeAvailabilityZone": "nova",
  "daFlavor": "dsr.da",
  "daImage": "DSR-9.0.2.0.0.vmdk",
  "daAvailabilityZone": "nova",
  "stpFlavor": "dsr.stp",
  "stpImage": "DSR-9.0.2.0.0.vmdk",
  "stpAvailabilityZone": "nova",
  "soamAffinityPolicy": "anti-affinity",
  "ipfeAffinityPolicy": "anti-affinity",
  "daAffinityPolicy": "soft-anti-affinity",
  "stpAffinityPolicy": "soft-anti-affinity"
}
}

```

Sample Request for signaling flavor DIAMETER + STP with serviceMp, multiple xsi(4 xsi interface) for DYNAMIX IP(Single Subnet - ipv4) deployment model.

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate`

Accept: application or json

Content type: application or json

Instantiating the first signaling VNF request for Fixed IP deployment (VNFM-V1)

Request Json:

Signaling flavor DIAMETER + STP with serviceMp,multiple xsi(4 xsi interface) for DYNAMIX IP(Single Subnet - ipv4) (VNFM - V1)

```

{
    "flavourId": "DIAMETER+STP",

```

```

        "instantiationLevelId": "small",
        "extVirtualLinks": "extVirtualLinks",
        "extManagedVirtualLinks": [{
            "id": "",
            "virtualLinkDescId": "active
NOAM",
            "resourceId":
"8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
        },
        {
            "id": "",
            "virtualLinkDescId": "standby
NOAM",
            "resourceId":
"2bed5886-8c97-4623-8da3-9c500cce71e3"
        }
    ],
    "vimConnectionInfo": {
        "MvlCloud": {
            "vimId": "vimId",
            "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
            "interfaceInfo": {
                "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
            },
            "accessInfo": {
                "project": "*****",
                "region": "RegionOne",
                "username": "*****",
                "password": "*****",
                "userDomain": "Default",
                "projectDomain": "default"
            }
        }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
        "xmiNetwork": {
            "name": "ext-net3",
            "subnet": [{
                "ipVersion": "IPv6",
                "name": "ext-net-ipv6-subnet"
            }]
        },
        "imiNetwork": {
            "name": "imi-private",
            "subnet": [{
                "ipVersion": "IPv6",
                "name": "test6"
            }]
        },
        "xsiNetwork": [{
            "name": "ext-net2",
            "subnet": [{
                "ipVersion": "IPv6",
                "name": "xsiIPv6"
            }]
        }
    ],

```

```

{
  "name": "xsiNetworkDual2",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "xsiNetworkDual2-IPv6"
  }]
},
  "ntpServerIp": "10.250.32.10",
  "primaryNoamVmName": "NOAM00-32cd6138",
  "noamSgName":
"dsrNetworkOam_NOAM_32cd6138_SG",
  "serviceMp": "true",
  "soamFlavor": "dsr.soam",
  "soamImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
  "soamAvailabilityZone": "nova",
  "ipfeFlavor": "dsr.ipfe",
  "ipfeImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
  "ipfeAvailabilityZone": "nova",
  "daFlavor": "dsr.da",
  "daImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
  "daAvailabilityZone": "nova",
  "stpFlavor": "dsr.stp",
  "stpImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
  "stpAvailabilityZone": "nova",
  "serviceMpFlavor": "dsr.ServiceMp",
  "serviceMpImage": "DSR-9.0.2.0.0_99.8.0.vmdk",
  "serviceMpAvailabilityZone": "nova",
  "soamAffinityPolicy": "anti-affinity",
  "ipfeAffinityPolicy": "anti-affinity",
  "daAffinityPolicy": "soft-anti-affinity",
  "stpAffinityPolicy": "soft-anti-affinity"
}
}

```

vnfConfigurableProperties json:

Signaling flavor DIAMETER + STP with serviceMp, multiple xsi(4 xsi interface)
for DYNAMIX IP(Single Subnet - ipv4) (VNFM - V2)

```

{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 4,
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [{
      "ipVersion": "IPv4",
      "name": "imi-private-sub"
    }]
  },
  "primaryNoamVmName": "Noam--2fa79d66-DN000",
  "noamSgName": "Noam__2fa79d66_DN0SG",
  "noamXmiIps": ["10.75.189.178", "10.75.189.206"],
  "serviceMp": "true"
}

```

```

    }
  }

  {
    "flavourId": "DIAMETER+SBR",
    "instantiationLevelId": "small",
    "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [{
      "id": "id1",
      "virtualLinkDescId": "active NOAM",
      "resourceId": "d6be6053-78a9-437a-a139-4dc11792598a"
    },
    {
      "id": "id2",
      "virtualLinkDescId": "standby NOAM",
      "resourceId": "d6be6053-78a9-437a-a139-4dc11792598a"
    }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net4",
      "subnet": [{
        "name": "ext-net4-subnet",
        "ipVersion": "IPv4",
        "fixedIps": {
          "primarySoamXmiIp": "10.75.218.141",
          "secondarySoamXmiIp": "10.75.218.163",
          "soamVip": "10.75.218.97",
          "dampXmiIps": [
            "10.75.218.38",
            "10.75.218.137"
          ],
          "ipfeXmiIps": [
            "10.75.218.153",
            "10.75.218.126"
          ],
          "sbrXmiIps": [
            "10.75.218.67",

```

```

        "10.75.218.84",
        "10.75.218.184"
    ]
}
    ]
}
    ],
    "imiNetwork": {
        "name": "imi-private",
        "subnet": [{
            "name": "imi-private-sub",
            "ipVersion": "IPv4",
            "fixedIps": {
                "primarySoamImiIp": "192.167.2.1",
                "secondarySoamImiIp": "192.167.2.2",
                "dampImiIps": [
                    "192.167.2.3",
                    "192.167.2.4"
                ],
                "ipfeImiIps": [
                    "192.167.2.5",
                    "192.167.2.6"
                ],
                "sbrImiIps": [
                    "192.167.2.7",
                    "192.167.2.8",
                    "192.167.2.9"
                ]
            }
        ]
    }
},
    "sbrNetwork": {
        "name": "ext-net7",
        "subnet": [{
            "name": "ext-net7-subnet",
            "ipVersion": "IPv4",
            "fixedIps": {
                "sbrNetworkIps": [
                    "10.196.218.95",
                    "10.196.218.108",
                    "10.196.218.18"
                ]
            }
        ]
    }
},
    "xsiNetwork": [{
        "name": "ext-net4",
        "subnet": [{
            "name": "ext-net4-subnet",
            "ipVersion": "IPv4",
            "fixedIps": {
                "dampXsiIps": [
                    "10.75.218.140",
                    "10.75.218.155"
                ],
                "ipfeXsiIps": [

```

```

        "10.75.218.101",
        "10.75.218.22"
    ]
    }
}
},
{
    "name": "ext-net4",
    "subnet": [{
        "name": "ext-net-sub",
        "ipVersion": "IPv4",
        "fixedIps": {
            "dampXsiIps": [
                "10.75.218.42",
                "10.75.218.122"
            ],
            "ipfeXsiIps": [
                "10.75.218.91",
                "10.75.218.131"
            ]
        }
    ]
}
}
},
    "ntpServerIp": "10.250.32.10",
    "primaryNoamVmName": "NOAM00-f1888e6d",
    "noamSgName": "dsrNetwork0am_NOAM_f1888e6d_SG",
    "soamFlavor": "dsr.soam",
    "soamImage": "DSR-9.0.2.0.0.vmdk",
    "soamAvailabilityZone": "nova",
    "ipfeFlavor": "dsr.ipfe",
    "ipfeImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeAvailabilityZone": "nova",
    "daFlavor": "dsr.da",
    "daImage": "DSR-9.0.2.0.0.vmdk",
    "daAvailabilityZone": "nova",
    "sbrFlavor": "dsr.sbr",
    "sbrImage": "DSR-9.0.2.0.0.vmdk",
    "sbrAvailabilityZone": "nova",
    "sbrAffinityPolicy": "affinity",
    "soamAffinityPolicy": "anti-affinity",
    "ipfeAffinityPolicy": "anti-affinity",
    "daAffinityPolicy": "soft-anti-affinity"
}
}
}

```

Sample Response

202 Accepted

Headers:

```

{
    location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
    date: Tue, 29 Jan 2019 10:39:24 GMT
    content-length: 0    content-type:

```

```

    application/xml
}

```

Sample Request

Instantiating the signaling VNF request with SBR (DIAMETER+SBR) with multiple xsi (1, 2, or 4 xsi interface) generated for Single Subnet Dynamic IP deployment model.

Resource URL: `https://<myhost-IP>:8443/vnflcm/<vnfm_version>/vnf_instances/<VNF ID received from create request>/instantiate`

Accept: application/json

Content-Type: application/json

Authorization: Token generated after login

Request JSON (VNFM - V1)

```

{
  "flavourId": "DIAMETER+SBR",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "subnet": [{
        "name": "ext-net3-subnet",
        "ipVersion": "IPv4"
      }]
    }
  }
}

```

```

    },
    "imiNetwork": {
      "name": "imi-private",
      "subnet": [{
        "name": "imi-private-sub",
        "ipVersion": "IPv4"
      }]
    },
    "xsiNetwork": [{
      "name": "ext-net2",
      "subnet": [{
        "name": "ext-net2-subnet",
        "ipVersion": "IPv4"
      }]
    }],
    {
      "name": "ext-net5",
      "subnet": [{
        "name": "ext-net5-subnet",
        "ipVersion": "IPv4"
      }]
    }
  ],
  "sbrNetwork": {
    "name": "ext-net3",
    "subnet": [{
      "name": "ext-net3-subnet",
      "ipVersion": "IPv4"
    }]
  },
  "ntpServerIp": "10.250.32.10",
  "primaryNoamVmName": "NOAM00-32cd6138",
  "noamSgName": "dsrNetwork0am_NOAM_32cd6138_SG",
  "soamFlavor": "dsr.soam",
  "soamImage": "DSR-9.0.2.0.0.vmdk",
  "soamAvailabilityZone": "nova",
  "ipfeFlavor": "dsr.ipfe",
  "ipfeImage": "DSR-9.0.2.0.0.vmdk",
  "ipfeAvailabilityZone": "nova",
  "daFlavor": "dsr.da",
  "daImage": "DSR-9.0.2.0.0.vmdk",
  "daAvailabilityZone": "nova",
  "sbrFlavor": "dsr.sbr",
  "sbrImage": "DSR-9.0.2.0.0.vmdk",
  "sbrAvailabilityZone": "nova",
  "sbrAffinityPolicy": "anti-affinity",
  "soamAffinityPolicy": "anti-affinity",
  "ipfeAffinityPolicy": "anti-affinity",
  "daAffinityPolicy": "soft-anti-affinity"
}
}

```

Instantiating the signaling VNF request with SBR (DIAMETER+SBR) with multiple xsi (1,2,4 xsi interface) generated for Fixed IP deployment model.

```
{
  "flavourId": "DIAMETER+SBR",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [
    {
      "id": "id1",
      "virtualLinkDescId": "active NOAM",
      "resourceId": "d6be6053-78a9-437a-a139-4dc11792598a"
    },
    {
      "id": "id2",
      "virtualLinkDescId": "standby NOAM",
      "resourceId": "d6be6053-78a9-437a-a139-4dc11792598a"
    }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      }
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default"
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net4",
      "subnet": [
        {
          "name": "ext-net4-subnet",
          "ipVersion": "IPv4",
          "fixedIps": {
            "primarySoamXmiIp": "10.75.218.141",
            "secondarySoamXmiIp": "10.75.218.163",
            "soamVip": "10.75.218.97",
            "dampXmiIps": [
              "10.75.218.38",
              "10.75.218.137"
            ]
          },
          "ipfexmiIps": [
            "10.75.218.153",
            "10.75.218.126"
          ],
          "sbrXmiIps": [
            "10.75.218.67",
            "10.75.218.84",

```

```

        "10.75.218.184"
    ]
}
}]
},
"imiNetwork": {
    "name": "imi-private",
    "subnet": [{
        "name": "imi-private-sub",
        "ipVersion": "IPv4",
        "fixedIps": {
            "primarySoamImiIp": "192.167.2.1",
            "secondarySoamImiIp": "192.167.2.2",
            "dampImiIps": [
                "192.167.2.3",
                "192.167.2.4"
            ],
            "ipfeImiIps": [
                "192.167.2.5",
                "192.167.2.6"
            ],
            "sbrImiIps": [
                "192.167.2.7",
                "192.167.2.8",
                "192.167.2.9"
            ]
        }
    ]
}
}]
},
"sbrNetwork": {
    "name": "ext-net7",
    "subnet": [{
        "name": "ext-net7-subnet",
        "ipVersion": "IPv4",
        "fixedIps": {
            "sbrNetworkIps": [
                "10.196.218.95",
                "10.196.218.108",
                "10.196.218.18"
            ]
        }
    ]
}
}]
},
"xsiNetwork": [{
    "name": "ext-net4",
    "subnet": [{
        "name": "ext-net4-subnet",
        "ipVersion": "IPv4",
        "fixedIps": {
            "dampXsiIps": [
                "10.75.218.140",
                "10.75.218.155"
            ],
            "ipfeXsiIps": [
                "10.75.218.101",

```

```

        "10.75.218.22"
    ]
  }
}
},
{
  "name": "ext-net4",
  "subnet": [{
    "name": "ext-net-sub",
    "ipVersion": "IPv4",
    "fixedIps": {
      "dampXsiIps": [
        "10.75.218.42",
        "10.75.218.122"
      ],
      "ipfeXsiIps": [
        "10.75.218.91",
        "10.75.218.131"
      ]
    }
  ]
}
}
},
"ntpServerIp": "10.250.32.10",
"primaryNoamVmName": "NOAM00-f1888e6d",
"noamSgName": "dsrNetworkOam_NOAM_f1888e6d_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0.vmdk",
"soamAvailabilityZone": "nova",
"ipfeFlavor": "dsr.ipfe",
"ipfeImage": "DSR-9.0.2.0.0.vmdk",
"ipfeAvailabilityZone": "nova",
"daFlavor": "dsr.da",
"daImage": "DSR-9.0.2.0.0.vmdk",
"daAvailabilityZone": "nova",
"sbrFlavor": "dsr.sbr",
"sbrImage": "DSR-9.0.2.0.0.vmdk",
"sbrAvailabilityZone": "nova",
"sbrAffinityPolicy": "affinity",
"soamAffinityPolicy": "anti-affinity",
"ipfeAffinityPolicy": "anti-affinity",
"daAffinityPolicy": "soft-anti-affinity"
}
}
}

```

For signaling flavors with UDR with multiple xsi (1 and 2 XSI interface) for Fixed IP deployment model

```

{
  "flavourId": "DIAMETER+UDR",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "id1",
    "virtualLinkDescId": "active NOAM",

```

```

        "resourceId": "6ba09324-0568-4489-bdb6-bcc9bb6218a3"
    },
    {
        "id": "id2",
        "virtualLinkDescId": "standby NOAM",
        "resourceId": "379e4fce-61a7-4323-8ee3-d548e819042f"
    }
],
"vimConnectionInfo": {
    "MvlCloud": {
        "vimId": "vimId",
        "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
        "interfaceInfo": {
            "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "project": "*****",
            "region": "RegionOne",
            "username": "*****",
            "password": "*****",
            "userDomain": "Default",
            "projectDomain": "default"
        }
    }
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net4",
        "subnet": [{
            "name": "ext-net4-subnet",
            "ipVersion": "IPv4",
            "fixedIps": {
                "primarySoamXmiIp": "10.75.218.207",
                "secondarySoamXmiIp": "10.75.218.218",
                "soamVip": "10.75.218.204",
                "primaryUdrXmiIp": "10.75.218.243",
                "secondaryUdrXmiIp": "10.75.218.223",
                "udrVip": "10.75.218.191",
                "dampXmiIps": [
                    "10.75.218.196",
                    "10.75.218.213"
                ],
                "ipfeXmiIps": [
                    "10.75.218.226",
                    "10.75.218.216"
                ]
            }
        ]
    }
}]
},

"imiNetwork": {
    "name": "imi-private",
    "subnet": [{
        "name": "imi-private-sub",

```

```

        "ipVersion": "IPv4",
        "fixedIps": {
            "primarySoamImiIp": "192.167.2.1",
            "secondarySoamImiIp": "192.167.2.2",
            "primaryUdrImiIp": "192.167.2.3",
            "secondaryUdrImiIp": "192.167.2.4",
            "dampImiIps": [
                "192.167.2.5",
                "192.167.2.6"
            ],
            "ipfeImiIps": [
                "192.167.2.7",
                "192.167.2.8"
            ]
        }
    },
    "xsiNetwork": [{
        "name": "ext-net4",
        "subnet": [{
            "name": "ext-net4-subnet",
            "ipVersion": "IPv4",
            "fixedIps": {
                "dampXsiIps": [
                    "10.75.218.214",
                    "10.75.218.217"
                ],
                "ipfeXsiIps": [
                    "10.75.218.149",
                    "10.75.218.238"
                ],
                "primaryUdrXsiIps": [
                    "10.75.218.201"
                ],
                "secondaryUdrXsiIps": [
                    "10.75.218.215"
                ]
            }
        ]
    }],
    {
        "name": "ext-net4",
        "subnet": [{
            "name": "ext-net4-subnet",
            "ipVersion": "IPv4",
            "fixedIps": {
                "dampXsiIps": [
                    "10.75.218.235",
                    "10.75.218.178"
                ],
                "ipfeXsiIps": [
                    "10.75.218.225",
                    "10.75.218.219"
                ],
                "primaryUdrXsiIps": [

```

```

        "10.75.218.175"
      ],
      "secondaryUdrXsiIps": [
        "10.75.218.230"
      ]
    }
  }
}
},
"ntpServerIp": "10.250.32.10",
"primaryNoamVmName": "NOAM00-a2eaba59",
"noamSgName": "dsrNetworkOam_NOAM_a2eaba59_SG",

"soamFlavor": "dsr.soam",

"soamImage": "DSR-9.0.2.0.0.vmdk",

"soamAvailabilityZone": "nova",

"ipfeFlavor": "dsr.ipfe",

"ipfeImage": "DSR-9.0.2.0.0.vmdk",

"ipfeAvailabilityZone": "nova",

"daFlavor": "dsr.da",

"daImage": "DSR-9.0.2.0.0.vmdk",

"daAvailabilityZone": "nova",

"udrFlavor": "udr.noam",

"udrImage": "UDR-12.6.1.0.0_18.7.0.vmdk",
"udrType": "vMNP",
"udrAvailabilityZone": "nova",
"soamAffinityPolicy": "anti-affinity",
"ipfeAffinityPolicy": "anti-affinity",
"daAffinityPolicy": "soft-anti-affinity",
"udrAffinityPolicy": "anti-affinity"
}
}

```

Sample Response

Instantiating the signaling VNF with SBR response

202 Accepted

Headers:

location: https:// <<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/

vnf_lcm_op_occs/lcmOp-f00678f4-ea8e-417f-9c5a-e126926db402

date: Wed, 13 Feb 2019 09:55:01 GMT

content-length: 0

content-type: application/xml

The following table describes the rules and required custom size values of DSR Signaling flavor:

Table 10-14 DSR Signaling Flavor Custom Size Values

DSR Signaling Flavor	Rules	Values
DIAMETER	numDa and numIpfe are mandatory fields	Minimum value of numDa to be passed should be 1 Minimum value of numIpfe to be passed should be 2
STP	numStp is a mandatory field	Minimum value of numStp to be passed should be 1
DIAMETER+STP	numDa and numStp are mandatory fields	Minimum value of numDa to be passed should be 1 Minimum value of numStp to be passed should be 1
DIAMETER+SBR	numSbr is a mandatory field	Minimum value of numSbr to be passed should be 3
DIAMETER+UDR	numUdr is a mandatory field	Minimum value of numUdr to be passed should be 2
STP+UDR	numUdr is a mandatory field	Minimum value of numUdr to be passed should be 2

- Damp can be maximum 32 and minimum 1.
- Stp can be maximum 32 and minimum 1.
- Ipfe can be maximum 4. It exists in pair and minimum is 2.
- Sbr can be maximum 9. It exists in multiple of 3 and minimum is 3.
- Udr can be maximum 4. It exists in pair and minimum is 2.
- Total C-level server count should not exceed 48.

Note

- The 202 response means that the request was accepted for processing. The VNF might take up to 15 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.
- If the VNFM creates a VNF that is operational, but has no DIAMETER configuration data, then create the necessary configuration data (peers, connections, etc.) to perform DIAMETER routing.
- The flavor ID must be selected based on the VMs to be deployed and the instantiation level must be selected based on the number of VMs required.
- Only the IPs of the required VM must be provided in the `fixedIp` parameter.

For Example

`"flavorId": "DIAMETER+STP", "instantiationLevelId": "small"` - This brings up 2 SOAM, 2 DAMP, 2 IPFE, 2 STP servers.

- The user must provide `primarySoamXmiIp(1)`, `secondarySoamXmiIp(1)`, `soamVip(1)`, `dampXmiIps(2)`, `ipfeXmiIps(2)`, `stpXmiIps(2)`, `dampXsiIps(2)`, `ipfeXsiIps(2)`, `stpXsiIps(2)`
- DA-MP proceeds for Cloud-init after SOAM installation is complete.

TSA configurations (for Signaling flavors DIAMETER, DIAMETER+SBR, DIAMETER+STP with multiple XSI (1,2,3,4 xsi))

Target set Address (TSA) is actually used for load balancing of the traffic. So IPFE will route all the incoming traffic and in return traffic will not route through IPFE but directly through gateway.

For configuring TSA, two things need to be configured on active Soam (Soam Vip):-

Detailed explanation of XMI, IMI and XSI Network

- IPFE optionSets
- IPFE target Sets

Note

Please use DSR-8.4.0.6.0-89.1.1 or higher version to enable this feature.

The detailed explanation of XMI and XSI Network for the additional parameters are provided below:

IPFE Option Sets Configuration:-

There are four parameters for configuring OptionSets i.e `IpfeA1IpAddress`, `IpfeA2IpAddress`, `IpfeB1IpAddress` and `IpfeB2IpAddress`. These options accept the IMI IP address of IPFE servers.

These are just logical names that facilitate binding with IPFE server. "OptionSets" parameter in request body is not mandatory. So by default VNFM configures OptionSets in following way:-

- For Single subnet IMI network and single pair of IPFE Servers (2 IPFE servers):
`IpfeA1IpAddress` will bind to `ipfe00`
`IpfeA2IpAddress` will bind to `ipfe01`

IpfeB1IpAddress and IpfeB2IpAddress will not be configured.

- For Single subnet IMI network and two pair of ipfe Servers (4 ipfe servers):
IpfeA1IpAddress will bind to ipfe00
IpfeA2IpAddress will bind to ipfe01
IpfeB1IpAddress will bind to ipfe02
IpfeB2IpAddress will bind to ipfe03
- For Dual subnet IMI network and single pair of Ipfe Servers(2 ipfe servers):
IpfeA1IpAddress will bind to ipfe00 (IPv4 address)
IpfeA2IpAddress will bind to ipfe01 (IPv4 address)
IpfeB1IpAddress will bind to ipfe00 (IPv6 address)
IpfeB2IpAddress will bind to ipfe01 (IPv6 address)
- For Dual subnet IMI network and two pair of Ipfe Servers(4 ipfe servers)
IpfeA1IpAddress will bind to ipfe00 (IPv4 address)
IpfeA2IpAddress will bind to ipfe01 (IPv4 address)
IpfeB1IpAddress will bind to ipfe00 (IPv4 address)
IpfeB2IpAddress will bind to ipfe01 (IPv4 address)

So to change the default configurations, pass the "OptionSets" in request body and configurations will be done accordingly.

IPFE Target Sets Configuration:-

Target sets can be configured for a particular XSI network. So pass the tsa parameter in xsi network. Multiple tsa can be configured for single xsi network. "tsa" configuration has following parameters:-

- dampName (optional in tsa) - By default VNFM will add all Damp's to all TSA's. But if only specific damp's need to be configured then it can be passed in this.
- fixedIp (optional in tsa) - By default VNFM will create port with dynamic IP based on xsi Network. If fixed ip needs to be configured then it can be passed in this.
- preferredActiveIPFE (mandatory in tsa) - It is based on OptionSets configurations. If IpfeA1IpAddress is configured then use ipfeA1 here, for IpfeA2IpAddress use ipfeA2, for IpfeB1IpAddress use ipfeB1 and for IpfeB2IpAddress use ipfeB2
- preferredStandByIPFE (mandatory in tsa) - Similar to preferredActiveIPFE.

. So if xsi network is on single subnet then port for tsa will be created only for single ip address so tsa configuration will be done for single IP only.

If xsi Network is on dual subnet then port for tsa will be created with two ip address. Both ip address will be configured in tsa.

Note

TSA configuration is supported for different combination of DIAMETER flavor.

Note

- If instantiation of stack has TSA configuration and if scaling operation is done on the same stack then scaled Damp's will be added to all the TSA present.

Note

If scaling is performed after discovery operations, then scaled DAMP will not be added to the TSA. Because discovery operation will not be having information about TSA.

Sample Request for signaling flavour DIAMETER with TSA, 1 xsi for DYNAMIC IP(Single Subnet - Ipv4) deployment model.

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate`

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

For XMI Network

Signaling DIAMETER with TSA, 2 xsi for DYNAMIC IP(Single Subnet - Ipv4) (VNFM - V2)

```
{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 2,
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [{
      "ipVersion": "IPv4",
      "name": "imi-private-sub"
    }]
  },
  "primaryNoamVmName": "csarNoam--4f613a9f-DN000",
  "noamSgName": "csarNoam__4f613a9f_DN0SG",
  "noamXmiIps": ["10.75.234.121", "10.75.234.61"],
  "tsa": {
    "ipfeXsiNetwork1": [{
      "dampName": ["damp01", "damp00"],
      "preferredActiveIPFE": "IpfeA1",
      "preferredStandByIPFE": "IpfeA1"
    }],
    "ipfeXsiNetwork2": [{
      "dampName": ["damp01", "damp00"],
      "preferredActiveIPFE": "IpfeA1",
      "preferredStandByIPFE": "IpfeA1"
    }],
    {
      "dampName": ["damp01", "damp00"],
      "preferredActiveIPFE": "IpfeA1",
      "preferredStandByIPFE": "IpfeA1"
    }
  ]
}
```

```
    }
}
```

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

```
{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 1,
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [{
      "ipVersion": "IPv4",
      "name": "imi-private-sub",
      "fixedIps": {
        "primarySoamImiIp": "10.75.189.181",
        "secondarySoamImiIp": "10.75.189.182",
        "dampImiIps": [
          "10.75.189.185",
          "10.75.189.186"
        ],
        "ipfeImiIps": [
          "10.75.189.187",
          "10.75.189.188"
        ]
      }
    ]
  },
  "primaryNoamVmName": "csarNoam--4f613a9f-DN000",
  "noamSgName": "csarNoam__4f613a9f_DN0SG",
  "noamXmiIps": ["10.75.234.121", "10.75.234.61"],
  "tsa": {
    "ipfeXsiNetwork1": [{
      "dampName": ["damp00", "damp01"],
      "preferedActiveIPFE": "IpfeA1",
      "preferedStandByIPFE": "IpfeA1",
      "fixedIp": ["10.75.189.231"]
    }]
  }
}
```

Sample Request for signaling flavour DIAMETER with TSA, 4 xsi for DYNAMIC IP(Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Request Json:

Signaling DIAMETER with TSA, 4 xsi for DYNAMIC IP(Dual Subnet) (VNFM - V1)

```
{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
], "vimConnectionInfo": {
  "MvlCloud": {
    "vimId": "vimId",
    "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
    "interfaceInfo": {
      "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default"
    }
  }
},
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "vipSubnetName": "ext-net-ipv6-subnet",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
      }],
      {
        "ipVersion": "IPv4",
        "name": "ext-net-ipv4-subnet"
      }
    ]
  },
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "test6"
    }
  ]
}
```

```

    },
    {
      "ipVersion": "IPv4",
      "name": "test4"
    }
  ]
},
"xsiNetwork": [{
  "name": "ext-net2",
  "tsa": [{
    "dampName": ["damp00", "damp01"],
    "preferedActiveIPFE": "IpfeA1",
    "preferedStandByIPFE": "IpfeA2",
    "fixedIp": ["", ""]
  },
  {
    "dampName": ["damp00", "damp01"],
    "preferedActiveIPFE": "IpfeA1",
    "preferedStandByIPFE": "IpfeA2",
    "fixedIp": ["", ""]
  }
],
"subnet": [{
  "ipVersion": "IPv6",
  "name": "xsiIPv6"
}],
{
  "ipVersion": "IPv4",
  "name": "xsiIPv4"
}
],
{
  "name": "xsiNetworkDual2",
  "tsa": [{
    "dampName": ["damp00", "damp01"],
    "preferedActiveIPFE": "IpfeA1",
    "preferedStandByIPFE": "IpfeA2",
    "fixedIp": ["", ""]
  },
  {
    "dampName": ["damp00", "damp01"],
    "preferedActiveIPFE": "IpfeA1",
    "preferedStandByIPFE": "IpfeA2",

```

```

        "fixedIp": ["", ""]
    },
    ],
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual2-IPv6"
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual2-IPv4"
    }
    ]
},
"OptionSets": {
    "ipfeA1IpAddress": ["ipfe00", "IPv6"],
    "ipfeA2IpAddress": ["ipfe01", "IPv6"],
    "ipfeB1IpAddress": ["ipfe02", "IPv6"],
    "ipfeB2IpAddress": ["ipfe03", "IPv6"]
},
"ntpServerIp": "10.250.32.10",
"primaryNoamVmName": "NOAM00-32cd6138",
"noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0.vmdk",
"soamAvailabilityZone": "nova",
"ipfeFlavor": "dsr.ipfe",
"ipfeImage": "DSR-9.0.2.0.0.vmdk",
"ipfeAvailabilityZone": "nova",
"daFlavor": "dsr.da",
"daImage": "DSR-9.0.2.0.0.vmdk",
"daAvailabilityZone": "nova",
"soamAffinityPolicy": "anti-affinity",
"ipfeAffinityPolicy": "anti-affinity",
"daAffinityPolicy": "soft-anti-affinity"
}
}

```

vnfConfigurableProperties Json:

Signaling DIAMETER with TSA, 1 xsi for FIXED IP(Single Subnet - Ipv4) (VNFM - V2)

```

{
    "ntpServerIp": "10.250.32.10",
    "xsiNetworkCount": 1,
    "imiNetwork": {
        "name": "imi-private",
        "subnet": [{
            "ipVersion": "IPv4",
            "name": "imi-private-sub",
            "fixedIps": {
                "primarySoamImiIp": "10.75.189.181",
                "secondarySoamImiIp": "10.75.189.182",
            }
        }
    ]
}

```

```

        "dampImiIps": [
            "10.75.189.185",
            "10.75.189.186"
        ],
        "ipfeImiIps": [
            "10.75.189.187",
            "10.75.189.188"
        ]
    }
}
}],
"primaryNoamVmName": "csarNoam--4f613a9f-DN000",
"noamSgName": "csarNoam_4f613a9f_DN0SG",
"noamXmiIps": ["10.75.234.121", "10.75.234.61"],
"tsa": {
    "ipfeXsiNetwork1": [{
        "dampName": ["damp00", "damp01"],
        "preferedActiveIPFE": "IpfeA1",
        "preferedStandByIPFE": "IpfeA1",
        "fixedIp": ["10.75.189.231"]
    }]
}
}
}

```

Sample Request for signaling flavour DIAMETER with TSA, 4 xsi for FIXED IP(Dual Subnet) deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

vnfConfigurableProperties Json:

Signaling DIAMETER with TSA, 4 xsi for FIXED IP(Dual Subnet) (VNFM - V2)

```

{
  "ntpServerIp": "10.250.32.10",
  "xsiNetworkCount": 4,
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "name": "test11",
      "ipVersion": "IPv4",
      "fixedIps": {
        "primarySoamImiIp": "10.75.189.170",
        "secondarySoamImiIp": "10.75.189.171",
        "dampImiIps": [
          "10.75.189.172",
          "10.75.189.173"
        ],
        "ipfeImiIps": [
          "10.75.189.174",
          "10.75.189.175"
        ]
      }
    }]
  }
}

```

```

    ]
  },
  {
    "name": "test6",
    "ipVersion": "IPv6",
    "fixedIps": {
      "primarySoamImiIp": "2606:b400:605:b818:6e41:6aff:fec7:77d0",
      "secondarySoamImiIp": "2606:b400:605:b818:6e41:6aff:fec7:76d0",
      "dampImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:75d0",
        "2606:b400:605:b818:6e41:6aff:fec7:74d0"
      ],
      "ipfeImiIps": [
        "2606:b400:605:b818:6e41:6aff:fec7:73d0",
        "2606:b400:605:b818:6e41:6aff:fec7:72d0"
      ]
    }
  }
],
  "primaryNoamVmName": "FT--19b44716-DN000",
  "noamSgName": "FT__19b44716_DNOSG",
  "noamXmiIps": ["10.75.234.51", "10.75.234.64"],
  "tsa": {
    "ipfeXsiNetwork1": [{
      "dampName": ["damp00", "damp01"],
      "preferedActiveIPFE": "IpfeA1",
      "preferedStandByIPFE": "IpfeA1",
      "fixedIp": ["10.75.189.231", "2606:b400:605:b818:6e41:6aff:fec7:836c"]
    }, {
      "dampName": ["damp00", "damp01"],
      "preferedActiveIPFE": "IpfeA1",
      "preferedStandByIPFE": "IpfeA1",
      "fixedIp": ["10.75.189.231", "2606:b400:605:b818:6e41:6aff:fec7:836c"]
    }, {
      "dampName": ["damp00", "damp01"],
      "preferedActiveIPFE": "IpfeA1",
      "preferedStandByIPFE": "IpfeA1",
      "fixedIp": ["10.75.189.231", "2606:b400:605:b818:6e41:6aff:fec7:836c"]
    }, {
      "dampName": ["damp00", "damp01"],
      "preferedActiveIPFE": "IpfeA1",
      "preferedStandByIPFE": "IpfeA1",
      "fixedIp": ["10.75.189.231", "2606:b400:605:b818:6e41:6aff:fec7:836c"]
    }
  ]
}
}

```

Sample Response

Instantiating the signaling VNF with TSA Configuration with DIAMETER response

```
202 Accepted Headers: location: https:// <>:8443/vnflcm/<>/vnf_lcm_op_occs/
lcmOp-f00678f4-ea8e-417f-9c5a-e126926db402
    date: Wed, 13 Feb 2019 09:55:01 GMT content-length: 0 content-type:
    application/xml
```

Note

User can use the NOAM XMI IP for "ntpServerIp" field while creating the dsr Signaling stack if and only if noam is synced with NTP. The following table describes the parameters used for sending request to VNFM:

Parameters and Definitions for Dynamic IP deployment model**Table 10-15 Parameters and Definitions for Dynamic IP deployment model VNFM-V1**

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavour to be instantiated
instantiationLevelId	Identifier of the instantiation level of the deployment flavour to be instantiated. If not present, the default instantiation level as declared in the VNFD is instantiated.
resourceId	The identifier of the resource - active NOAM and then standBy NOAM (in that order) - in the scope of the VIM or the resource provider or value of resourceId can be provided as IP's of active NOAM and then standby NOAM (in that order)
xmiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for internal communication
name	Network name, for example; ext-net
ipVersion	IP version IPv4 or IPv6
xsiNetwork	Network used for DSR signaling traffic
ntpServerIp	IP of the NTP server
primaryNoamVmName	Name of primary NOAM VM on which the config XML is loaded
noamSgName	The server group of the NOAM VM
soamFlavor (optional)	flavor used for openstack deploys
soamImage (optional)	image used for openstack deploys
soamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
ipfeFlavor (optional)	flavor used for openstack deploys
ipfeImage (optional)	image used for openstack deploys (If using MultiQueue feature, Please follow the Prerequisite for the IPFE image)
ipfeAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
daFlavor (optional)	flavor used for openstack deploys

Table 10-15 (Cont.) Parameters and Definitions for Dynamic IP deployment model VNFM-V1

Parameter	Definitions
dalmage (optional)	image used for openstack deploys
daAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
stpFlavor (optional)	flavor used for openstack deploys
stpImage (optional)	image used for openstack deploys
stpAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
sbrFlavor (optional)	name of logical partitioning in case of host aggregate
sbrImage (optional)	image used for openstack deploys
sbrAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
udrFlavor (optional)	flavor used for openstack deploys
udrImage (optional)	image used for openstack deploys
udrAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
vipSubnetName (In case of Dual Subnet)	Name of VIP subnet to be used only in case of Dual Subnet
soamAffinityPolicy (optional)	openstack affinity policy for SOAM
daAffinityPolicy (optional)	openstack affinity policy for DAMP
ipfeAffinityPolicy (optional)	openstack affinity policy for IPFE
sbrAffinityPolicy (optional)	openstack affinity policy for SBR
stpAffinityPolicy (optional)	openstack affinity policy for STP
udrAffinityPolicy (optional)	openstack affinity policy for UDR
udrType (optional)	Type of udr to be created
tsa (optional)	for configuring tsa
optionSets (optional)	for configuring optionSets of tsa
numDa (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Damp Server
numIpfe (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Ipfe Server
numStp (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Stp Server
numServiceMp (Optional - Used only during "custom" instantiation level id based on "STP" flavor)	Number of ServiceMp Server
numUdr (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Udr Server
numSbr (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Sbr Server

Note

The flavor ID must be chosen based on the VMs to be deployed. Then the instantiation level is to be chosen based on the number of VMs required.

Only the IPs of the required VM are to be passed in the fixedIp parameter.

For Example: "flavorId": "DIAMETER+STP", "instantiationLevelId": "small" -> This will bring up 2 SOAM, 2 DAMP, 2 IPFE, 2 STP servers. (Please refer to the above table on flavor and instantiation level for more info)

The user needs to pass primarySoamXmiIp(1), secondarySoamXmiIp(1), soamVip(1), dampXmiIps(2), ipfeXmiIps(2), stpXmiIps(2), dampXsiIps(2), ipfeXsiIps(2), stpXsiIps(2)

Here is the detailed explanation of XMI Network in the additional parameters of the request body:

Detailed Explanation for XMI Network

```
"xmiNetwork":{
  "name":"<NAME of the network of XMI IPS >",
  "subnet":[{
    "name": "<Name of the Subnet of XMI network>"
    "ipVersion":"IPv4",
    "fixedIps":{
      "primarySoamXmiIp":"<ACTIVE SOAM XMI IP>",
      "secondarySoamXmiIp":"<STANDBY SOAM XMI IP>",
      "soamVip":"<SOAM VIP>",
      "dampXmiIps":[
        "<DAMP 00 XMI IP>",
        "<DAMP 01 XMI IP>"
      ],
      "ipfeXmiIps":[
        "<IPFE 00 XMI IP>",
        "<IPFE 01 XMI IP>"
      ],
      "stpXmiIps":[
        "<STP 00 XMI IP>",
        "<STP 01 XMI IP>"
      ]
    }
  ]
}
```

For IMI Network

```
"xmiNetwork":{
  "name":"<NAME of the network of XMI IPS >",
  "subnet":[{
    "name": "<Name of the Subnet of XMI Network>",
    "ipVersion":"IPv4",
    "fixedIps":{
      "primarySoamImiIp":"<ACTIVE SOAM IMI IP>",
      "secondarySoamImiIp":"<STANDBY SOAM IMI IP>",
      "dampImiIps":[
        "<DAMP 00 IMI IP>",
        "<DAMP 01 IMI IP>"
      ],
      "ipfeImiIps":[
        "<IPFE 00 IMI IP>",
        "<IPFE 01 IMI IP>"
      ]
    }
  ]
}
```

```

    ],
    "stpImiIps": [
        "<STP 00 IMI IP>",
        "<STP 01 IMI IP>"
    ]
  }
}
}

```

For XSI Network

```

"xsiNetwork": [
  {
    "name": "<NAME of the network of XSI 1>",
    "subnet": [ {
      "name": "<Name of the Subnet of XSI-1 network>",
      "ipVersion": "IPv4",
      "fixedIps": {
        "dampXsiIps": [
          "<DAMP00 XSI 1 IP>",
          "<DAMP 01 XSI 1 IP>"
        ],
        "ipfeXsiIps": [
          "<IPFE00 XSI 1 IP>",
          "<IPFE01 XSI 1 IP>"
        ]
      },
      "stpXsiIps": [
        "<STP00 XSI 1 IP>",
        "<STP01 XSI 1 IP>"
      ]
    }
  ],
  {
    "name": "<NAME of the network of XSI 2>",
    "subnet": [ {
      "name": "<Name of the Subnet of XSI-2 network>",
      "ipVersion": "IPv4",
      "fixedIps": {
        "dampXsiIps": [
          "<DAMP00 XSI 2 IP>",
          "<DAMP01 XSI 2 IP>"
        ],
        "ipfeXsiIps": [
          "<IPFE00 XSI 2 IP>",
          "<IPFE01 XSI 2 IP>"
        ]
      },
      "stpXsiIps": [
        "<STP00 XSI 2 IP>",
        "<STP01 XSI 2 IP>"
      ]
    }
  ]
}
]

```

The following describes the parameters used for sending request to VNFM.

Table 10-16 Parameters and Definitions for Signaling VNF with Multiple XSI

Parameters	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated
instantiationLevelId	Identifier of the instantiation level of the deployment flavor to be instantiated. If not present, the default instantiation level as declared in the VNFD is instantiated.
resourceId	The identifier of the resource (active NOAM and then standBy NOAM) in the scope of the VIM or the resource provider
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for internal communication
name	Network name, for example; ext-net
ipVersion	IP version IPv4 or IPv6
xsiNetwork	Network that is used for DSR signaling traffic
ntpServerIP	IP of the NTP server
primaryNoamVmName	Name of primary NOAM VM on which the configured XML is loaded
noamSgName	The server group of the NOAM VM
primarySoamXmiIp	IP address of primary SOAM
secondarySoamXmiIp	IP address of secondary SOAM
soamVip	VIP of SOAM
dampXmiIps	List of DAMP external management IPs (only if DAMPs are being instantiated)
ipfeXmiIps	List of IPFE external management IPs (only if IPFEs are being instantiated)
stpXmiIps	List of vSTP external management IPs (only if STPs are being instantiated)
dampXsiIps	List of DAMP signaling IPs (only if DAMPs are being instantiated)
ipfeXsiIps	List of IPFE signaling IPs (only if IPFEs are being instantiated)
stpXsiIps	List of STP signaling IPs (only if STPs are being instantiated)
primaryUdrXmiIp	IP address of primary UDR (only if UDRs are being instantiated)
secondaryUdrXmiIp	IP address of secondary UDR (only if UDRs are being instantiated)
udrVip	VIP address of UDR (only if UDRs are being instantiated)
primaryUdrXsiIps	List of primary UDR signaling IPs (only if UDRs are being instantiated)
secondaryUdrXsiIps	List of secondary UDR signaling IPs (only if UDRs are being instantiated)

Table 10-16 (Cont.) Parameters and Definitions for Signaling VNF with Multiple XSI

Parameters	Definitions
sbrXmiIps	List of SBR external management IPs (only if SBRs are being instantiated)
sbrNetworkIps	List of SBR replication port IPs (only if SBRs are being instantiated)
primarySoamImiIp	IP address of primary SOAM for IMI
secondarySoamImiIp	IP address of secondary SOAM for IMI
dampImiIps	List of DAMP internal management IPs (only if DAMPs are being instantiated)
ipfeImiIps	List of IPFE internal management IPs (only if IPFEs are being instantiated)
stpImiIps	List of vSTP internal management IPs (only if STPs are being instantiated)
primaryUdrImiIp	IP address of primary UDR for IMI (only if UDRs are being instantiated)
secondaryUdrImiIp	IP address of secondary UDR for IMI (only if UDRs are being instantiated)
sbrImiIps	List of SBR internal management IPs (only if SBRs are being instantiated)
soamFlavor (optional)	flavor used for OpenStack deploys
soamImage (optional)	image used for OpenStack deploys
soamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
ipfeFlavor (optional)	flavor used for OpenStack deploys
ipfeImage (optional)	image used for OpenStack deploys If you are using the Multiqueue feature, you must follow the prerequisite for the IPFE image.
ipfeAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
daFlavor (optional)	flavor used for OpenStack deploys
daImage (optional)	image used for OpenStack deploys
daAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
stpFlavor (optional)	flavor used for OpenStack deploys
stpImage (optional)	image used for OpenStack deploys
stpAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
sbrFlavor (optional)	flavor used for OpenStack deploys
sbrImage (optional)	image used for OpenStack deploys
sbrAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
udrFlavor (optional)	flavor used for OpenStack deploys
udrImage (optional)	image used for OpenStack deploys
udrAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
vipSubnetName (In case of Dual Subnet)	name of VIP subnet to be used only in case of Dual Subnet

Table 10-16 (Cont.) Parameters and Definitions for Signaling VNF with Multiple XSI

Parameters	Definitions
soamAffinityPolicy (optional)	openstack affinity policy for SOAM
daAffinityPolicy (optional)	openstack affinity policy for DAMP
ipfeAffinityPolicy (optional)	openstack affinity policy for IPFE
sbrAffinityPolicy (optional)	openstack affinity policy for SBR
stpAffinityPolicy (optional)	openstack affinity policy for STP
udrAffinityPolicy (optional)	openstack affinity policy for UDR
udrType (optional)	Type of UDR to be created
tsa (optional)	for configuring tsa
optionSets (optional)	for configuring optionSets of tsa
numDa (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Damp Server
numIpfe (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Ipfe Server
numStp (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Stp Server
numUdr (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Udr Server
numSbr (Optional - Used only during "custom" instantiation level id based on flavor)	Number of Sbr Server

10.10.2 Instantiating the Signaling VNF with Service MP deployment

VNFM deploys the Service MP along with vSTP-MP only. Service MP deployment is optional. vSTP-MP can be independently deployed without service MP.

To enable Service MP Deployment, update **"serviceMp": "true"** in the Signaling request. Default value is "false".

If Service MP is not included (**"serviceMp": "false"**) while vSTP-MP instantiation, the Service MP can be scaled up from 0 to any number (till the maximum supported size) by scale using **"scaleInfo (Arbitrary Size)"** operation.

Sample Request for DIAMETER+STP for dynamic IP deployment model VNFM V1

This is a sample request for signaling flavors with Service MP - STP, DIAMETER+STP with multiple xsi (1,2,4 xsi interface) for dynamic IP deployment model.

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
  "flavourId": "DIAMETER+STP",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
      }]
    },
    "imiNetwork": {
      "name": "imi-private",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6"
      }]
    },
    "xsiNetwork": [{
      "name": "ext-net2",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiIPv6"
      }]
    }
  ],
  {

```

```

        "name": "xsiNetworkDual2",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "xsiNetworkDual2-IPv6"
        }]
    },
    "ntpServerIp": "10.250.32.10",
    "primaryNoamVmName": "NOAM00-32cd6138",
    "noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
    "soamFlavor": "dsr.soam",
    "soamImage": "DSR-9.0.2.0.0.vmdk",
    "soamAvailabilityZone": "nova",
    "ipfeFlavor": "dsr.ipfe",
    "ipfeImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeAvailabilityZone": "nova",
    "daFlavor": "dsr.da",
    "daImage": "DSR-9.0.2.0.0.vmdk",
    "daAvailabilityZone": "nova",
    "stpFlavor": "dsr.stp",
    "stpImage": "DSR-9.0.2.0.0.vmdk",
    "stpAvailabilityZone": "nova",
    "soamAffinityPolicy": "anti-affinity",
    "ipfeAffinityPolicy": "anti-affinity",
    "daAffinityPolicy": "soft-anti-affinity",
    "stpAffinityPolicy": "soft-anti-affinity"
}
}

```

Sample Request for DIAMETER+STP for Fixed IP deployment model.

Sample Request for signaling flavors without SBR. DIAMETER,STP and DIAMETER+STP, DIAMETER+UDR, STP+UDR) with multiple xsi (1,2,4 xsi interface) for Fixed IP deployment model.

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate`

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
  "flavourId": "DIAMETER+STP",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [
    {
      "id": "id1",
      "virtualLinkDescId": "",
      "resourceId": "d6be6053-78a9-437a-a139-4dc11792598a"
    },
    {
      "id": "id2",
      "virtualLinkDescId": "",
      "resourceId": "d6be6053-78a9-437a-a139-4dc11792598a"
    }
  ]
}

```

```

    }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default",
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net4",
      "subnet": [
        {
          "ipVersion": "IPv4",
          "name": "ext-net-subnet",
          "fixedIps": {
            "primarySoamXmiIp": "10.75.218.141",
            "secondarySoamXmiIp": "10.75.218.163",
            "soamVip": "10.75.218.97",
            "dampXmiIps": [
              "10.75.218.38",
              "10.75.218.137"
            ],
            "ipfeXmiIps": [
              "10.75.218.153",
              "10.75.218.126"
            ],
            "stpXmiIps": [
              "10.75.218.67",
              "10.75.218.84"
            ],
            "serviceMpXmiIps": [
              "10.75.218.69",
              "10.75.218.85"
            ]
          }
        }
      ]
    }
  }
},
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [
      {
        "name": "imi-private-sub",
        "ipVersion": "IPv4",
        "fixedIps": {

```

```

        "primarySoamImiIp": "192.167.2.9",
        "secondarySoamImiIp": "192.167.2.10",
        "dampImiIps": [
            "192.167.2.11",
            "192.167.2.12"
        ],
        "ipfeImiIps": [
            "192.167.2.13",
            "192.167.2.14"
        ],
        "stpImiIps": [
            "192.167.2.15",
            "192.167.2.16"
        ],
        "serviceMpImiIps": [
            "192.167.2.18",
            "192.167.2.19"
        ]
    }
}]]
},
    "xsiNetwork": [
        {
            "name": "ext-net4",
            "subnet": [
                {
                    "name": "ext-net4-subnet",
                    "ipVersion": "IPv4",
                    "fixedIps": {
                        "dampXsiIps": [
                            "10.75.218.140",
                            "10.75.218.155"
                        ],
                        "ipfeXsiIps": [
                            "10.75.218.101",
                            "10.75.218.22"
                        ],
                        "stpXsiIps": [
                            "10.75.218.95",
                            "10.75.218.108"
                        ]
                    }
                }
            ]
        }
    ]
},
{
    "name": "ext-net",
    "subnet": [
        {
            "name": "ext-net-subnet",
            "ipVersion": "IPv4",
            "fixedIps": {
                "dampXsiIps": [
                    "10.75.218.140",
                    "10.75.218.155"
                ],
            }
        }
    ]
}

```

```

        "ipfeXsiIps": [
            "10.75.218.101",
            "10.75.218.22"
        ],
        "stpXsiIps": [
            "10.75.218.95",
            "10.75.218.108"
        ]
    }
}

}],
    "ntpServerIp": "10.250.32.10",
    "primaryNoamVmName": "NOAM00-",
    "noamSgName": "dsrNetworkOam_NOAM__SG",
    "serviceMp": "true",
    "soamFlavor": "dsr.soam",
    "soamImage": "DSR-9.0.2.0.0.vmdk",
    "soamAvailabilityZone": "nova",
    "ipfeFlavor": "dsr.ipfe",
    "ipfeImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeAvailabilityZone": "nova",
    "daFlavor": "dsr.da",
    "daImage": "DSR-9.0.2.0.0.vmdk",
    "daAvailabilityZone": "nova",
    "stpFlavor": "dsr.stp",
    "stpImage": "DSR-9.0.2.0.0.vmdk",
    "stpAvailabilityZone": "nova",
    "soamAffinityPolicy": "anti-affinity",
    "ipfeAffinityPolicy": "anti-affinity",
    "daAffinityPolicy": "soft-anti-affinity",
    "stpAffinityPolicy": "soft-anti-affinity"
}
}

```

10.11 Instantiating Multiple Signaling VNFs

To instantiate multiple Signaling VNFs, simply repeat the above procedures. You would need to create another DSR Signaling VNF instance, and you must deploy each Signaling VNF on a separate OpenStack instance.

Note

For lab installations, a separate tenant on the same OpenStack instance is acceptable.

10.12 Instantiating the IDIH VNF

Supports dynamic and fixed IP deployment model.

To start IDIH deployment, it is required to instantiate a signaling VNF. Before deploying the VNF, ensure the following information is available:

- The VNF ID for a previously created IDIH VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
 - OpenStack Controller URI
 - User Domain Name
 - Project Domain Id
 - Username
 - Password
 - Tenant name
- The name of a public network in the selected OpenStack instance that carries the IDIH traffic.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server, and is often a good choice.
- The network ID of the IMI private network of the signaling stack to carry OAM traffic in the selected OpenStack instance. A signaling VNF is a prerequisite as the ID of the internal network generated from this stack must be used for instantiating IDIH.
- The name of the internal private network in the selected OpenStack instance that will allow communication between Application, Mediation, and Database servers.

For more information about the full list of all inputs and possible outputs of the `instantiate` VNF command, see **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the DSR VNFM Swagger specification.

Sample request for dynamic IP deployment model

Instantiating IDIH Request generated

URL: `https://<<VNFM HOST IP>>:8443/vnfm/v1/vnf_instances/< VNF ID received from create request>/instantiate`

Accept: `application/json`

Content-Type: `application/json`

Authorization : `<Token generated after login>`

```
{
  "flavourId":"IDIH",
  "instantiationLevelId":"small",
  "extVirtualLinks":"extVirtualLinks",

  "extManagedVirtualLinks":[

    {
      "id":"id1",
      "virtualLinkDescId":"Network ID of the network used for intra-site
communication(iml) with Signalling VNF",

      "resourceId":"aae72b3d-d189-4464-a217-58bb0320065b"
    }
  ],
}
```

```

    "vimConnectionInfo": {
      "MvlCloud": {
        "vimId": "vimId",
        "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
        "interfaceInfo": {
          "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        },
        "accessInfo": {
          "project": "*****",
          "region": "RegionOne",
          "username": "*****",
          "password": "*****",
          "userDomain": "Default",
          "projectDomain": "default"
        }
      }
    },

    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
      "ntpServerIp": "10.250.32.10",
      "xmiNetwork": {
        "name": "ext-net3",

        "ipVersion": "IPv4",

        "xmiSubnetName" : "ext-net3-subnet"
      },
      "idihIntNetwork": {
        "idihIntPrivateNetwork": "test",
        "idihIntPrivateSubnet": "test-sub"
      }
    },

    "idihAppFlavor": "appl-idih",

    "idihMedFlavor": "med-idih",

    "idihDbFlavor": "db-idih",

    "idihAppImage": "apps-8.2.3.0.0_82.38.1.vmdk",

    "idihMedImage": "mediation-8.2.3.0.0_82.38.1.vmdk",

    "idihDbImage": "oracle-8.2.3.0.0_82.38.1.vmdk",

    "idihAvailabilityZone": "nova",
    "idihKeyName": "idih-test-key"
  }
}

```

Sample Request for fixed IP deployment model

Instantiating IDIH Request for fixed IP model generated
 URL: https://<<VNFM HOST IP>>:8443/vnfm/v1/vnf_instances/< VNF ID received

```

from create request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>
{
  "flavourId":"IDIH",
  "instantiationLevelId":"small",
  "extVirtualLinks":"extVirtualLinks",

  "extManagedVirtualLinks":[

    {
      "id":"id1",
      "virtualLinkDescId":" Network ID of the network used for intra-site
communication(iml) with Signalling VNF",
      "resourceId":"aae72b3d-d189-4464-a217-58bb0320065b"
    }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "****",
        "password": "*****",
        "userDomain": "*****",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage":"localizationLanguage",
  "additionalParams":{
    "ntpServerIp":"10.250.32.10",
    "xmiNetwork":{
      "name":"ext-net3",

      "ipVersion":"IPv4",

      "xmiSubnetName" : "ext-net3-subnet",

      "fixedIps":{
        "idihDbXmiIp":"10.75.218.30",
        "idihMedXmiIp":"10.75.218.19",
        "idihAppXmiIp":"10.75.218.49"
      }
    }
  }
}

```

```

    },
    "idihIntNetwork":{
        "idihIntPrivateNetwork":"test",
        "idihIntPrivateSubnet":"test-sub"
    },

    "idihAppFlavor": "appl-idih",

    "idihMedFlavor": "med-idih",

    "idihDbFlavor": "db-idih",

    "idihAppImage": "apps-8.2.3.0.0_82.38.1.vmdk",

    "idihMedImage": "mediation-8.2.3.0.0_82.38.1.vmdk",

    "idihDbImage": "oracle-8.2.3.0.0_82.38.1.vmdk",

    "idihAvailabilityZone": "nova",

    "idihKeyName":"idih-test-key"

    }
}

```

10.13 Instantiating the SDS Network OAM VNF

SDS NOAM is a setup of following three servers:

- Primary Noam
- Secondary Noam
- Query Server

In order to start a SDS deployment, it is required to instantiate a SDS Network OAM VNF. Before deploying the VNF, the following information must be available:

- The VNF ID for a previously created SDS network OAM VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
 - OpenStack Controller URI
 - User Domain Name
 - Project Domain Id

- Username
- Password
- Project name
- The name of a public network in the selected OpenStack instance that will carry the OAM traffic.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server, and is often a good choice.
 - Supported for IPv6 networks - ipVersion should be "IPv6" in the request Body. The GUI can be accessed by the following URL: `https://[<SDS-NOAM-vIP>]`
For example: `https://[fd0d:deba:d97c:2c:6e41:6aff:fec7:80bf]`

For more information about the full list of all inputs and possible outputs of the **instantiate VNF** command, see **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the DSR VNFM Swagger specification .

Expected Alarms:

- 31226 - HA Availability Status Degraded (Major Alarm)
- 10012 - Table change responder failed (Major Alarm)
- 14101 - No Remote Connections (Major Alarm)
- 10073 - Server Group Max Allowed HA Role Warning (Minor Alarm)

Sample Request: For DYNAMIC IP deployment model

Instantiating SDS NOAM Request for Dynamic IP deployment model

Resource URL: `https://<VNFM HOST IP>:8443/vnflcm/<vnfm_version>/vnf_instances/<VNF ID received from create request>/instantiate`

Accept: `application/json`

Content-Type: `application/json`

Authorization: `<Token generated after login>`

```
{
  "flavourId": "SDS NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],

  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      }
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",

```

```

        "projectDomain": "default",
    }
}],
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net3",
        "vipSubnetName": "ext6-net3-subnet",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "ext6-net3-subnet"
        }],
        {
            "ipVersion": "IPv4",
            "name": "ext-net3-subnet"
        }
    ]
},
"imiNetwork": {
    "name": "imi-net3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "imi6-net3-subnet"
    }],
    {
        "ipVersion": "IPv4",
        "name": "imi-net3-subnet"
    }
}
],
"ntpServerIp": "10.250.32.10",
"sdsNoamFlavor": "sds.noam",
"sdsQsFlavor": "sds.noam",
"sdsNoamImage": "SDS-9.0.2.0.0.vmdk",
"sdsQsImage": "SDS-9.0.2.0.0.vmdk",
"sdsNoamAvailabilityZone": "nova",
"sdsQsAvailabilityZone": "nova",
"sdsNoamAffinityPolicy": "anti-affinity"
}
}

```

Note

The "vipSubnetName" field is used only in case of Dual Subnet.

Sample Request: For Fixed IP deployment model

Instantiating SDS NOAM Request For fixed IP deployment model.

Resource URL: https://<<VNFM_HOST_IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/<VNF_ID_received_from_create_request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization: <Token generated after login>

```
{
  "flavourId": "SDS NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],

  "vimConnectionInfo": {
    "MvlCloud": {
      "vimid": "vimid",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default",
    },
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net8",
      "vipSubnetName": "ext6-net3-subnet",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext6-net3-subnet",
        "fixedIps": {
          "primarySdsNoamIp": "2606:b400:605:b813::14",
          "secondarySdsNoamIp": "2606:b400:605:b813::13",
          "sdsQsIp": "2606:b400:605:b813::12",
          "sdsNoamVip": "2606:b400:605:b813::11"
        }
      }
    },
    {
      "ipVersion": "IPv4",
      "name": "ext-net3-subnet",
      "fixedIps": {
        "primarySdsNoamIp": "10.75.218.50",
        "secondarySdsNoamIp": "10.75.218.49",
        "sdsQsIp": "10.75.218.134"
      }
    }
  },
  "imiNetwork": {
    "name": "imi-net",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "imi6-net-subnet",
      "fixedIps": {
        "primarySdsNoamImiIp": "2606:b400:605:b813:12",

```

```

        "secondarySdsNoamImiIp": "2606:b400:605:b813::1",
        "sdsQsImiIp": "2606:b400:605:b813::14"
    },
    {
        "ipVersion": "IPv4",
        "name": "imi-net-subnet",
        "fixedIps": {
            "primarySdsNoamImiIp": "192.167.2.5",
            "secondarySdsNoamImiIp": "192.167.2.4",
            "sdsQsImiIp": "192.167.2.3"
        }
    }
}

},
"ntpServerIp": "10.250.32.10",
"sdsNoamFlavor": "sds.noam",
"sdsQsFlavor": "sds.noam",
"sdsNoamImage": "SDS-9.0.2.0.0.vmdk",
"sdsQsImage": "SDS-9.0.2.0.0.vmdk",
"sdsNoamAvailabilityZone": "nova",
"sdsQsAvailabilityZone": "nova",
"sdsNoamAffinityPolicy": "anti-affinity"
}
}

```

Sample Response

Instantiating NOAM Request

202 Accepted

Headers:

```

{
    location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
    date: Tue, 29 Jan 2019 10:39:24 GMT
    content-length: 0   content-type:
    application/xml
}

```

Note

- The 202 response means that the request was accepted for processing. The VNF might take up to 15 minutes to become fully operational. Use the SDS GUI to determine when the VNF is operational.
- After SDS NOAM VNF deployment, standby SDS NOAM is automatically changed to **Force StandBy**, purposely to avoid any switchover while SDS Signaling VNF is deployed. After SDS Signaling site is deployed and no more Life Cycle Management operations are planned, make **Force Standby** NOAM as **Active** by changing the **Max Allowed HA Role** to **Active** on **Status & Manage** -> **HA** from **Active SDS NOAM GUI**.
- The supported SDS NOAM Flavor is SDS NOAM.
- The supported SDS NOAM Flavor instantiation level id is HA that creates 2 SDS NOAMs and 1 Query Server.

The following table describes the parameters used for sending request to VNFM:

Table 10-17 Parameters and Definitions

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavour to be instantiated.
xmiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for inter-site communication.
imiNetwork	Network used for internal communication of DSR entities
ntpServerIp	IP of the NTP server
sdsNoamFlavor (optional)	flavor used for openstack deploys
sdsQsFlavor (optional)	flavor used for openstack deploys
sdsNoamImage (optional)	image used for openstack deploys
sdsQsImage (optional)	image used for openstack deploys
sdsNoamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
sdsQsAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
sdsNoamAffinityPolicy (optional)	openstack affinity policy for SDS NOAM

Table 10-18 Parameters and Definitions for Fixed IP deployment

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used for internal communication of DSR entities
ntpServerIp	IP of the NTP server

Table 10-18 (Cont.) Parameters and Definitions for Fixed IP deployment

Parameter	Definitions
fixedIps	Json object in network to provide IP address
primarySdsNoamIp	IP address for primary SDS NOAM IP
secondarySdsNoamIp	IP address for secondary SDS NOAM IP
sdsQsIp	IP address for SDS Query Server VIP
sdsNoamVip	IP address for SDS NOAM VIP
primarySdsNoamImiIp	IP address for primary SDS NOAM IP of IMI
secondarySdsNoamImiIp	IP address for secondary SDS NOAM IP of IMI
sdsQsImiIp	IP address for SDS Query Server IP of IMI
sdsNoamFlavor (optional)	flavor used for OpenStack deploys
sdsQsFlavor (optional)	flavor used for OpenStack deploys
sdsNoamImage (optional)	image used for OpenStack deploys
sdsQsImage (optional)	image used for OpenStack deploys
sdsNoamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate for sdsNoam
sdsQsAvailabilityZone (optional)	name of logical partitioning in case of host aggregate for sdsQs
sdsNoamAffinityPolicy (optional)	openstack affinity policy for SDS NOAM

10.14 Instantiating the SDS DR Network OAM VNF

SDS DRNOAM is the Disaster Recovery SDS NOAM site. In case both the Active and Standby SDS NOAM of Primary site fails, then the operator can make SDS DRNOAM as the Primary Site and can continue the operations without any disturbance.

When a setup is configured with a SDS DR NOAM then the first SDS NOAM SG is treated as the Primary NOAM Site and the second SDS NOAM SG is treated as Secondary NOAM site.

SDS DR NOAM is a setup of three servers:

- Primary Noam
- Secondary Noam
- Query Server

In order to instantiate a SDS DR Network OAM VNF, the following information must be available:

- The VNF ID for a previously created SDS DR network OAM VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
 - OpenStack Controller URI
 - User Domain Name
 - Project Domain Id
 - Username
 - Password
 - Project name

- The name of a public network in the selected OpenStack instance that will carry the OAM traffic.
- OpenStack resource IDs for the XMI IPs from both SDS NOAM VMs.

Note

The resource IDs can be obtained by examining the SDS Network OAM stack to which the identified SDS DR NOAM VNF is attached.

- Name of Active Primary SDS NOAM VM.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server, and is often a good choice.
- DSR DR NOAM supports Dual Subnet for XMI and IMI interfaces.

10.14.1 Determining the SDS DR NOAM XMI Resource IDs

The following facts must be considered before proceeding with SDS DR NOAM site creation:

- SDS DRNOAM site must be created on separate tenant.
- SDS DRNOAM site is referred to as Secondary NOAM. Therefore, we have two sites, Primary and Secondary. Secondary Site configuration is done on Primary Active SDS NOAM.
- In the Primary Active SDS NOAM, when second SDS NOAM Server Group gets created, it automatically becomes Secondary.
- The Primary Active SDS NOAM communicates to the Secondary Active SDS NOAM through existing Comcol replication and merging mechanism.
- The Secondary SDS NOAM Site is optional and does not require to be deployed at the same time as of the Primary SDS NOAM.

From the OpenStack GUI:

1. Change your view to the tenant on which the DSR Network OAM VNF was deployed.
2. Go to **Project->Network->Network Topology**. A diagram of all VMs in the tenant is displayed.

Note

The diagram may take a few minutes to display.

3. Click on one of the NOAM VMs.
4. A pop-up appears having information about the specific NOAM VM.
5. Save the resource ID for the XMI port provided in the IP Addresses section of the pop-up.

Note

The IP Addresses section of the popup contains information about the network ports and resource IDs, assigned to the VM.

6. Repeat the previous step for the other NOAM VM.

You can also use the following alternative:

- Instead of passing resource IDs, user can use SDS-NOAM XMI IPs.
- User can pass Active SDS-NOAM's XMI IP to resource id 1 and StandBy SDS-NOAM's XMI IP to resource id 2.

For more information about the full list of all inputs and possible outputs of the **instantiate VNF** command, see **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the DSR VNFM Swagger specification. Swagger specifications can be found post VNFM installation at (https://<VNFM_IP>:8443/docs/vnfm/).

Sample Request: Instantiating SDS DR NOAM Request for DYNAMIC IP deployment model

Resource URL: https://<VNFM_HOST_IP>:8443/vnflcm/<vnfm_version>/vnf_instances/<VNF_ID_received_from_create_request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization: Token generated after login

```
{
  "flavourId": "SDS DR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "id1",
    "virtualLinkDescId": "active SDS NOAM XMI",
    "resourceId": "156d73cf-6e44-456b-a661-14bd0cc2b43c"
  },
  {
    "id": "id2",
    "virtualLinkDescId": "standby SDS NOAM XMI",
    "resourceId": "5c638770-5585-44c7-97c7-b4a52a26e5ec"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "*****",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
```

```

"xmiNetwork": {
  "name": "ext-net3",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "ext6-net3-subnet"
  },
  {
    "ipVersion": "IPv4",
    "name": "ext-net3-subnet"
  }]
},
"imiNetwork": {
  "name": "imi-net",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "ext6-net3-subnet"
  },
  {
    "ipVersion": "IPv4",
    "name": "ext-net3-subnet"
  }]
},
"ntpServerIp": "10.250.32.10",
"primarySdsNoamVmName": "SDS-NOAM00-ea47f4b1",
"sdsDrNoamFlavor": "sds.noam",
"sdsDrQsFlavor": "sds.noam",
"sdsDrNoamImage": "SDS-9.0.2.0.0.vmdk",
"sdsDrQsImage": "SDS-9.0.2.0.0.vmdk",
"sdsDrNoamAvailabilityZone": "nova",
"sdsDrQsAvailabilityZone": "nova",
"sdsDrNoamAffinityPolicy": "anti-affinity"
}
}

```

Note

The "vipSubnetName" field is used only in case of Dual Subnet.

Instantiating SDS DR NOAM Request for Fixed IP deployment model

```

{
  "flavourId": "SDS DR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "id1",
    "virtualLinkDescId": "active SDS NOAM XMI",
    "resourceId": "156d73cf-6e44-456b-a661-14bd0cc2b43c"
  },
  {
    "id": "id2",
    "virtualLinkDescId": "standby SDS NOAM XMI",
    "resourceId": "5c638770-5585-44c7-97c7-b4a52a26e5ec"
  }
]
}

```

```

    },
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "*****",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext6-net3-subnet",
        "fixedIps": {
          "sdsDrPrimaryNoamIp": "2606:b400:605:b813::14",
          "sdsDrSecondaryNoamIp": "2606:b400:605:b813::13",
          "sdsDrQueryServerIp": "2606:b400:605:b813::12",
          "sdsDrNoamVip": "2606:b400:605:b813::11"
        }
      }
    },
    {
      "ipVersion": "IPv4",
      "name": "ext-net3-subnet",
      "fixedIps": {
        "sdsDrPrimaryNoamIp": "10.75.218.50",
        "sdsDrSecondaryNoamIp": "10.75.218.49",
        "sdsDrQueryServerIp": "10.75.218.134"
      }
    }
  ]
},
  "imiNetwork": {
    "name": "imi-net",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext6-net3-subnet",
      "fixedIps": {
        "sdsDrPrimaryNoamImiIp": "2606:b400:605:b813::14",
        "sdsDrSecondaryNoamImiIp": "2606:b400:605:b813::13",
        "sdsDrQueryServerImiIp":
"2606:b400:605:b813::12"
      }
    }
  ],
  {

```

```

        "ipVersion": "IPv4",
        "name": "ext-net3-subnet",
        "fixedIps": {
            "sdsDrPrimaryNoamImiIp": "10.75.218.50",
            "sdsDrSecondaryNoamImiIp": "10.75.218.49",
            "sdsDrQueryServerImiIp": "10.75.218.134"
        }
    }
}
},
"ntpServerIp": "10.250.32.10",
"primarySdsNoamVmName": "SDS-NOAM00-ea47f4b1",
"sdsDrNoamFlavor": "sds.noam",
"sdsDrQsFlavor": "sds.noam",
"sdsDrNoamImage": "SDS-9.0.2.0.0.vmdk",
"sdsDrQsImage": "SDS-9.0.2.0.0.vmdk",
"sdsDrNoamAvailabilityZone": "nova",
"sdsDrQsAvailabilityZone": "nova",
"sdsDrNoamAffinityPolicy": "anti-affinity"
}
}

```

Sample Response

202 Accepted

Headers:

```

{
    location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6

    date: Tue, 21 Feb 2019 10:39:24 GMT

    content-length: 0    content-type:

    application/xml
}

```

Note

- The 202 response means that the request was accepted for processing. The VNF might take up to 15 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.
- The supported SDS DR NOAM Flavor is SDS DR NOAM.
- The supported SDS DR NOAM Flavor instantiation level id is HA, which creates 2 SDS NOAMs and 1 Query Server.
- Supported for IPv6 networks - ipVersion should be "IPv6" in the request Body.

The following table describes the parameters used for sending request to VNFM:

Table 10-19 Parameters and Definitions SDS DR Network OAM VNF

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated
instantiationLevelId	Identifier of the instantiation level of the deployment flavor to be instantiated. If not present, the default instantiation level is HA.
resourceId	The identifier of the resource (active and then standby SDS NOAM XMI) in the scope of the VIM or the resource provider.
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used for internal communication of DSR entities
name	Network name, for example; ext-net
ipVersion	IP version IPv4 or IPv6
ntpServerIp	IP of the NTP server
primarySdsNoamVmName	Primary Active SDS NOAM VM name
sdsDrPrimaryNoamIp	XMI IP of the Primary SDS DR NOAM
sdsDrSecondaryNoamIp	XMI IP of the Secondary SDS DR NOAM
sdsDrQueryServerIp	XMI IP of the SDS DR QUERY NOAM
sdsDrNoamVip	VIP of the SDS DR NOAM
sdsDrPrimaryNoamImiIp	IMI IP of the Primary SDS DR NOAM
sdsDrSecondaryNoamImiIp	IMI IP of the Secondary SDS DR NOAM
sdsDrQueryServerImiIp	IMI IP of the Primary SDS DR NOAM
sdsDrNoamFlavor (optional)	flavor used for OpenStack deploys
sdsDrNoamImage (optional)	image used for OpenStack deploys
sdsDrQsFlavor (optional)	flavor used for OpenStack deploys
sdsDrQsImage (optional)	image used for OpenStack deploys
sdsDrNoamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
sdsDrQsAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
sdsDrNoamAffinityPolicy (optional)	openstack affinity policy for SDS DR NOAM

10.15 Instantiating the SDS Signaling VNF

In order to deploy the SDS signaling VNF, the following information must be available:

- A previously instantiated SDS network OAM VNF.
- The VNF ID for a previously created SDS signaling VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
 - OpenStack Controller URI
 - User Domain Name
 - Project Domain Id

- Username
- Password
- Tenant name
- The name of the xmi public network in the selected OpenStack instance that will carry traffic.
- The IP address of the NTP server accessible by VMs within the selected OpenStack instance.
- The OpenStack controller that controls the selected OpenStack instance normally hosts an NTP server, and is often a good choice.
- OpenStack resource IDs for the IMI IP from SDS Signaling and XMI IPs from both NOAM VMs.

Note

The resource IDs can be obtained by examining the SDS Network OAM stack and DSR Signaling stack to which the identified SDS signaling VNF would be attached.

- Name of the Active NOAM VM.

Note

To avoid switchover of Active NOAM, make the **StandBy** NOAM as **Forced Standby** by changing the **Max Allowed HA Role** to **Standby** on **Status & Manage** under **HA** from **Active NOAM GUI**.

- Name of the NOAM SG

Note

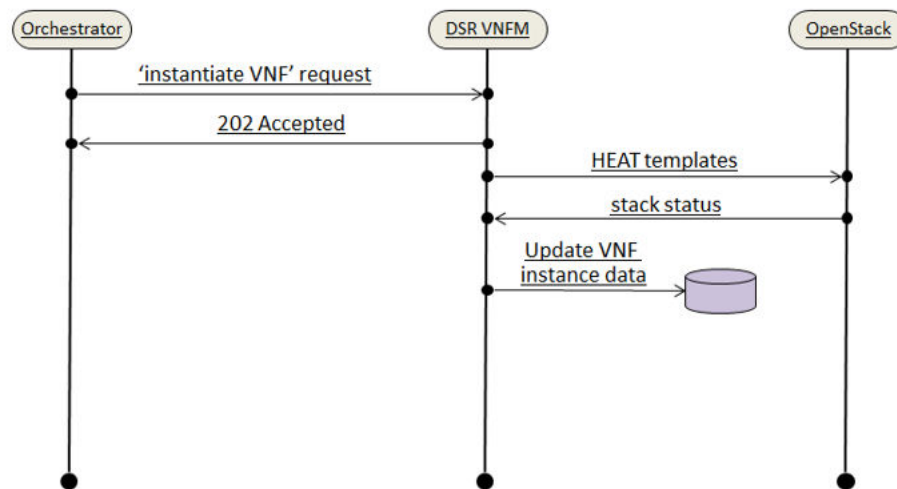
After SDS deployment, the Max Allowed HA Role of Query Server is expected to be Observer but it is Standby. Manually change the Max Allowed HA Role of Query Server from Standby to Observer as follows:

Log in to Active SDS NOAM GUI and navigate to **Status & Manage** under **HA** and click **Edit** to change the role of Query Server to Observer, and click **OK**.

- Supported for IPv6 networks - ipVersion should be "IPv6" in the request Body.
- SDS Signaling supports Dual IP

The following image illustrates the VNF instantiation:

Figure 10-6 SDS Signaling VNF Instantiate Request



The following table informs about the supported Instantiation levels to Instantiate VNF resource for SDS Signaling VNF:

Table 10-20 SDS Signaling Flavors supported by VNFM

Signaling Flavors supported by VNFM	Small	Medium	Large
	DP Server	DP Server	DP Server
SDSSIGNALING	2	6	10

Note

- The number of DP-SOAM is 2 for any instantiation level.
- SDS Signaling supports "custom" instantiation level id.

10.15.1 Determining the Signaling IMI Resource IDs

From the OpenStack GUI:

1. Navigate to **Project -> Network -> Networks**
2. Open the Network used for intra-site communication with Signaling VNF (imi).
3. The IMI resource ID is the ID of this network.

10.15.2 Determining the SDS NOAM XMI Resource IDs

From the OpenStack GUI:

- Change your view to the tenant on which the DSR Network OAM VNF is deployed.
- Go to **Project->Network->Network Topology**. A diagram of all VMs in the tenant is displayed.

Note

The diagram may take a few minutes to display.

- Click on one of the NOAM VMs.
- A pop-up appears having information about the specific NOAM VM.
- Save the resource ID for the XMI port provided in the IP Addresses section of the pop-up.

Note

The IP Addresses section of the popup contains information about the network ports and resource IDs, assigned to the VM.

- Repeat the previous step for the other NOAM VM and DSR Signaling VM.

You can also use the following alternative:

- Instead of passing resource IDs, user can use SDS-NOAM XMI IPs.
- User can pass Active SDS-NOAM's XMI IP to resource id 1 and StandBy SDS-NOAM's XMI IP to resource id 2.

Note

If SDS-NOAM is created on Dual Subnet then, then use IPv4 XMI IP's of SDS-NOAM while creating SDS-SOAM.

For more information about the full listing of all inputs and possible outputs of the command "instantiate VNF", see **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the DSR VNFM Swagger specification.

Sample Request:

Instantiating the first signaling VNF request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization: Token generated after login

Request JSON for Dynamic IP (VNFM - V1)

```
{
  "flavourId": "sdssignaling",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active SDS NOAM XMI",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }],
}
```

```

    {
      "id": "",
      "virtualLinkDescId": "standby SDS NOAM XMI",
      "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daeg3"
    }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "*****",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net3",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext6-net3-subnet",
        "fixedIps": {
          "primaryDpSoamXmiIp": "2606:b400:605:b813::11",
          "dpSoamXmiIp": "2606:b400:605:b813::11",
          "dpSoamVip": "2606:b400:605:b813::11",
          "dpXmiIps": ["2606:b400:605:b813::11"]
        }
      }],
      {
        "ipVersion": "IPv4",
        "name": "ext-net3-subnet",
        "fixedIps": {
          "primaryDpSoamXmiIp": "10.75.192.5",
          "dpSoamXmiIp": "10.75.192.6",
          "dpXmiIps": ["10.75.192.8"]
        }
      }
    }
  ],
  "imiNetwork": {
    "name": "imi-net3",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "imi6-net3-subnet",
      "fixedIps": {
        "primaryDpSoamImiIp": "2606:b400:605:b813::11",
        "dpSoamImiIp": "2606:b400:605:b813::11",
        "dpImiIps": ["2606:b400:605:b813::11"]
      }
    }
  ]
}

```

```

    }
  },
  {
    "ipVersion": "IPv4",
    "name": "imi-net3-subnet",
    "fixedIps": {
      "primaryDpSoamImiIp": "192.167.2.1",
      "dpSoamImiIp": "192.167.2.3",
      "dpImiIps": ["192.167.2.5"]
    }
  }
]
},
"ntpServerIp": "10.250.32.10",
"primarySdsNoamVmName": "SDS-NOAM00-32cd6138",
"sdsNoamSgName": "sdsNetworkOam_NOAM_32cd6138_SG",
"dpSoamFlavor": "sds.noam",
"dpFlavor": "sds.dpsoam",
"dpSoamImage": "SDS-9.0.2.0.0.vmdk",
"dpImage": "SDS-9.0.2.0.0.vmdk",
"dpSoamAvailabilityZone": "nova",
"dpAvailabilityZone": "nova",
"dpSoamAffinityPolicy": "anti-affinity",
"dpAffinityPolicy": "anti-affinity"
}
}
}

```

Note

The "vipSubnetName" field is used only in case of Dual Subnet.

Request JSON for Fixed IP (VNFM - V1)

```

{
  "flavourId": "sdssignaling",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active",
    "resourceId":
      "2bed5886-8c97-4623-8da3-9c500cce71e3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby",
    "resourceId":
      "8a4d1ec6-367a-4b1a-978d-2c4eae3daeg3"
  }
]
}

```

```

        },
        "vimConnectionInfo": {
            "MvlCloud": {
                "vimid": "vimid",
                "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
                "interfaceInfo": {
                    "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
                }
            },
            "accessInfo": {
                "username": "dsrsci.user",
                "password": "xxxx",
                "userDomain": "Default",
                "projectDomain": "default",
            }
        }
    },
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
        "xmiNetwork": {
            "name": "ext-net3",
            "subnet": [{
                "ipVersion": "IPv6",
                "name": "ext6-net3-
subnet",
                "fixedIps": {
                    "primaryDpSoamXmiIp": "2606:b400:605:b813::11",
                    "dpSoamXmiIp": "2606:b400:605:b813::11",
                    "dpSoamVip": "2606:b400:605:b813::11",
                    "dpXmiIps": ["2606:b400:605:b813::11"]
                }
            }
        },
        {
            "ipVersion": "IPv4",
            "name": "ext-net3-
subnet",
            "fixedIps": {
                "primaryDpSoamXmiIp": "10.75.192.5",
                "dpSoamXmiIp": "10.75.192.6",
                "dpXmiIps": ["10.75.192.8"]
            }
        }
    },
    "imiNetwork": {
        "name": "imi-net3",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "imi6-net3-
subnet",
            "fixedIps": {
                "primaryDpSoamImiIp": "2606:b400:605:b813::11",
                "dpSoamImiIp": "2606:b400:605:b813::11",
                "dpImiIps": ["2606:b400:605:b813::11"]
            }
        }
    },
    },

```

```

        {
            "ipVersion": "IPv4",
            "name": "imi-net3-
subnet",
            "fixedIps": {
                "primaryDpSoamImiIp": "192.167.2.1",
                "dpSoamImiIp": "192.167.2.3",
                "dpImiIps": ["192.167.2.5"]
            }
        }
    ]
},
    "ntpServerIp": "10.250.32.10",
    "primarySdsNoamVmName": "SDS-NOAM00-32cd6138",
    "sdsNoamSgName":
"sdsNetworkOam_NOAM_32cd6138_SG",
    "dpSoamFlavor": "sds.noam",
    "dpFlavor": "sds.dpsoam",
    "dpSoamImage": "SDS-9.0.2.0.0.vmdk",
    "dpImage": "SDS-9.0.2.0.0.vmdk",
    "dpSoamAvailabilityZone": "nova",
    "dpAvailabilityZone": "nova",
    "dpSoamAffinityPolicy": "anti-affinity",
    "dpAffinityPolicy": "anti-affinity"
}
}

```

Sample Response

202 Accepted

Headers:

```

{
    location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
    date: Tue, 29 Jan 2019 10:39:24 GMT
    content-length: 0    content-type:
    application/xml
}

```

Sample Request for Dynamic IP deployment with Custom Size model

Instantiating the SDS signaling with Custom Size VNF request generated

URL: https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/ < VNF
ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
    "flavourId": "sdssignaling",
    "instantiationLevelId": "custom",
    "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [{
        "id": "",

```

```

SDS NOAM XMI",
"2bed5886-8c97-4623-8da3-9c500cce71e3"
    },
    {
        "id": "",
        "virtualLinkDescId": "standby",
        "resourceId":

SDS NOAM XMI",
"8a4d1ec6-367a-4b1a-978d-2c4eae3daeg3"
    }
    ],
    "vimConnectionInfo": {
        "MvlCloud": {
            "vimid": "vimid",
            "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
            "interfaceInfo": {
                "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
            },
            "accessInfo": {
                "project": "*****",
                "region": "RegionOne",
                "username": "*****",
                "password": "*****",
                "userDomain": "Default",
                "projectDomain": "default",
            }
        }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
        "xmiNetwork": {
            "vipSubnetName": "ext6-net3-

subnet",
            "name": "ext-net3",
            "subnet": [{
                "ipVersion": "IPv6",
                "name": "ext6-net3-subnet"
            }],
            {
                "ipVersion": "IPv4",
                "name": "ext-net3-

subnet"
            }
        }
    },
    "imiNetwork": {
        "name": "imi-net",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "imi6-net-subnet"
        }],
        {
            "ipVersion": "IPv4",
            "name": "imi-net-

subnet"
        }
    }

```

```

    }
    "ntpServerIp": "10.250.32.10",
    "primarySdsNoamVmName": "SDS-NOAM00-32cd6138",
    "sdsNoamSgName":
"sdsNetworkOam_NOAM_32cd6138_SG",
    "dpSoamFlavor": "sds.noam",
    "dpFlavor": "sds.dpsoam",
    "dpSoamImage": "SDS-9.0.2.0.0.vmdk",
    "dpImage": "SDS-9.0.2.0.0.vmdk",
    "dpSoamAvailabilityZone": "nova",
    "dpAvailabilityZone": "nova",
    "dpSoamAffinityPolicy": "anti-affinity",
    "dpAffinityPolicy": "anti-affinity",
    "numDp": "2"
}
}

```

Sample Response

```

202 Accepted
Headers:
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0  content-type:
  application/xml
}

```

The following table describes the parameters used for sending request to VNFM:

Table 10-21 Parameters and Definitions for SDS Signaling VNF

Parameters	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated
instantiationLevelId	Identifier of the instantiation level of the deployment flavor to be instantiated. If not present, the default instantiation level as declared in the VNFD is instantiated.
resourceId	The identifier of the resource (imi Network ID of the signaling VNF, active, standby SDS NOAM XML) in the scope of the VIM or the resource provider
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for internal communication
name	Network name, for example; ext-net
ipVersion	IP version IPv4 or IPv6
ntpServerIp	IP of the NTP server

Table 10-21 (Cont.) Parameters and Definitions for SDS Signaling VNF

Parameters	Definitions
primarySdsNoamVmName	Name of primary SDS NOAM VM
sdsNoamSgName	The server group of the SDS NOAM VM
primaryDpSoamXmiIp	IP address for primary SDS DP SOAM IP
dpSoamXmiIp	IP address for secondary SDS DP SOAM IP
dpSoamVip	IP address for SDS SOAM VIP
dpXmiIps	IP address for SDS DP IP
primaryDpSoamImiIp	IP address for primary SDS DP SOAM IP of IMI
dpSoamImiIp	IP address for secondary SDS DP SOAM IP of IMI
dpImiIps	IP address for primary SDS DP IP of IMI
dpSoamFlavor (optional)	flavor used for openstack deploys
dpFlavor (optional)	flavor used for openstack deploys
dpSoamImage (optional)	image used for openstack deploys
dpImage (optional)	image used for openstack deploys
dpSoamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
dpAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
dpSoamAffinityPolicy (optional)	openstack affinity policy for SDS
SOAMdpAffinityPolicy (optional)	openstack affinity policy for SDS DP
numDp (Optional - Used only during "custom" instantiation level id)	Number of Dp Server. Minimum value is 1 and maximum is 10

Parameters and Definitions for Fixed IP Deployment model (VNFM - V1)

Table 10-22 Parameters and Definitions

Parameter	Definition
flavourId	Identifier of the VNF deployment flavour to be instantiated
instantiationLevelId	Identifier of the instantiation level of the deployment flavour to be instantiated. If not present, the default instantiation level as declared in the VNFD is instantiated.
resourceId	Value of resourceId can be provided as IP's of active SDS NOAM and then standby SDS NOAM (in that order)
xmiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
imiNetwork	Network used to provide access to the DSR entities (GUI, ssh), and for internal communication
name	Network name, for example; ext-net
ipVersion	IP version IPv4 or IPv6
ntpServerIp	IP of the NTP server
primarySdsNoamVmName	Name of primary SDS NOAM VM
sdsNoamSgName	The server group of the SDS NOAM VM

Table 10-22 (Cont.) Parameters and Definitions

Parameter	Definition
primaryDpSoamXmilp	IP address for primary SDS DP SOAM IP
dpSoamXmilp	IP address for secondary SDS DP SOAM IP
dpSoamVip	IP address for SDS SOAM VIP
dpXmilps	IP address for SDS DP IP
primaryDpSoamImilp	IP address for primary SDS DP SOAM IP of IMI
dpSoamImilp	IP address for secondary SDS DP SOAM IP of IMI
dplmilps	IP address for primary SDS DP IP of IMI
dpSoamFlavor (optional)	flavor used for openstack deploys
dpFlavor (optional)	flavor used for openstack deploys
dpSoamImage (optional)	mage used for openstack deploys
dplImage (optional)	image used for openstack deploys
dpSoamAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
dpAvailabilityZone (optional)	name of logical partitioning in case of host aggregate
dpSoamAffinityPolicy (optional)	openstack affinity policy for SDS
SOAMdpAffinityPolicy (optional)	openstack affinity policy for SDS DP
numDp (Optional - Used only during "custom" instantiation level id)	Number of Dp Server. Minimum value is 1 and maximum is 10.

10.16 Instantiating the ATS Master VNF (VNFM-V1)

The ATS Master VNF supports dynamic and fixed IP deployment models.

To deploy the ATS Master VNF, you must have the following pieces of information:

- The VNF ID for a previously created ATS Master VNF instance.
- Information about the OpenStack instance on which the VNF must be deployed:
 - OpenStack Controller URI
 - User Domain Name
 - Project Domain Id
 - Username
 - Password
 - Project name
- The name of a public network in the selected OpenStack instance that will carry the ATS master traffic.
- The IP of an NTP server accessible by VMs within the selected OpenStack instance. The OpenStack controller that controls the selected OpenStack instance hosts an NTP server.
- If you are using DNS, it must be accessible from instance.

For more information about the list of all the inputs and possible outputs of the command `instantiate VNF`, refer to **ETSI NFV-SOL 003**, section **5.4.4.3.1**, or the **DSR VNFM Swagger specification**.

Note

- ATS Master will always have 2 XSI while instantiating.
- Networks in ATS = XMI, XSI.
- Default and only instantiationLevelId is small
- Does not support Dual or mixed mode.

Sample Request for Dynamic IP Deployment Model

Request JSON for Fixed IP (VNFM - V1)

```
{
  "flavourId": "master",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "*****",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net8",
      "ipVersion": "IPv4",
      "xmiSubnetName": "ext-net8-subnet"
    },
    "xsiNetwork": [{
      "name": "ext-net7",
      "ipVersion": "IPv4",
      "xsiSubnetName": "ext-net7-subnet"
    },
    {
      "name": "ext-net6",
      "ipVersion": "IPv4",
      "xsiSubnetName": "ext-net6-subnet"
    }
  ],
  "ntpServerIp": "10.250.32.10",
  "dnsServerIp": "10.250.32.10",
  "atsKeyName": "atsKeypair",

```

```

        "atsMasterFlavor": "ats.master",
        "atsMasterImage": "ats-8.4.0.5.0-84.15.0",
        "atsAvailabilityZone": "nova"
    }
}

```

vnfConfigurableProperties Json:

Request JSON for Fixed IP (VNFM - V2)

```

{
    "ntpServerIp": "10.250.32.10",
    "dnsServerIp": "10.250.32.10",
    "atsKeyName": "atsKeypair"
}

```

Sample Request for Fixed IP Deployment Model

Request JSON for Fixed IP (VNFM - V1)

URL: `https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request>/instantiate`

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
    "flavourId": "master",
    "instantiationLevelId": "small",
    "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [],
    "vimConnectionInfo": {
        "MvlCloud": {
            "vimId": "vimId",
            "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
            "interfaceInfo": {
                "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
            }
        },
        "accessInfo": {
            "project": "*****",
            "region": "RegionOne",
            "username": "*****",
            "password": "*****",
            "userDomain": "*****",
            "projectDomain": "default"
        }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
        "xmiNetwork": {
            "name": "ext-net8",
            "ipVersion": "IPv4",
            "xmiSubnetName": "ext-net8-subnet",
            "fixedIps": {

```

```

        "masterXmiIp": "10.75.123.16"
    },
    "xsiNetwork": [{
        "name": "ext-net7",
        "ipVersion": "IPv4",
        "xsiSubnetName": "ext-net7-subnet",
        "fixedIps": {
            "xsiIp": "10.75.195.21"
        }
    },
    {
        "name": "ext-net6",
        "ipVersion": "IPv4",
        "xsiSubnetName": "ext-net6-subnet",
        "fixedIps": {
            "xsiIp": "10.75.195.22"
        }
    }
],
    "ntpServerIp": "10.250.32.10",
    "dnsServerIp": "10.250.32.10",
    "atsKeyName": "atsKeypair",
    "atsMasterFlavor": "ats.master",
    "atsMasterImage": "ats-8.4.0.5.0-84.15.0",
    "atsAvailabilityZone": "nova"
}
}

```

vnfConfigurableProperties Json:

Request JSON for Dynamic IP (VNFM - V2)

```

{
    "ntpServerIp": "10.250.32.10",
    "dnsServerIp": "10.250.32.10",
    "atsKeyName": "atsKeypair"
}

```

Sample Request for Fixed IP deployment model

URL: https://<>:8443/vnfm/<>/vnf_instances/< VNF ID received from create request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

Request JSON for Fixed IP (VNFM - V1)

```

{
    "flavourId": "master",
    "instantiationLevelId": "small",
    "extVirtualLinks": "extVirtualLinks",

```

```

"extManagedVirtualLinks": [],
"vimConnectionInfo": {
  "MvlCloud": {
    "vimId": "vimId",
    "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
    "interfaceInfo": {
      "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "*****",
      "projectDomain": "default"
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net8",
      "ipVersion": "IPv4",
      "xmiSubnetName": "ext-net8-subnet",
      "fixedIps": {
        "masterXmiIp": "10.75.123.16"
      }
    },
    "xsiNetwork": [{
      "name": "ext-net7",
      "ipVersion": "IPv4",
      "xsiSubnetName": "ext-net7-subnet",
      "fixedIps": {
        "xsiIp": "10.75.195.21"
      }
    },
    {
      "name": "ext-net6",
      "ipVersion": "IPv4",
      "xsiSubnetName": "ext-net6-subnet",
      "fixedIps": {
        "xsiIp": "10.75.195.22"
      }
    }
  ],
  "ntpServerIp": "10.250.32.10",
  "dnsServerIp": "10.250.32.10",
  "atsKeyName": "atsKeypair",
  "atsMasterFlavor": "ats.master",
  "atsMasterImage": "ats-8.4.0.5.0-84.15.0",
  "atsAvailabilityZone": "nova"
}
}

```

vnfConfigurableProperties Json:

Request JSON for Fixed IP (VNFM - V2)

```
{  "ntpServerIp": "10.250.32.10",  "dnsServerIp": "10.250.32.10",
   "atsKeyName": "atsKeypair"
}
```

Sample Response

```
{
  "ntpServerIp": "10.250.32.10",
  "dnsServerIp": "10.250.32.10",
  "atsKeyName": "atsKeypair"
}
```

Instantiating the ATS Master VNF response

202 Accepted

Headers:

```
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6

  date: Tue, 29 Jan 2019 10:39:24 GMT

  content-length: 0   content-type:

  application/xml
}
```

The following table describes the parameters for ATS Master (VNFM - V1) :

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavor to be instantiated.
xmiNetwork	Network used to provide access master VM communication.
ntpServerIp	IP of the NTP server.
dnsServerIp (optional)	IP of the DNS server. If not provided, NTP server IP will be considered as DNS server IP.
atsKeyName	Key pair name for ATS. To log in to the ATS instance, use same key pair.
masterXmiIp	In case of fixed IP scenario, the IP of master will be provided.
xsiNetwork	Network used for DSR signaling traffic.
atsMasterFlavor (optional)	Flavor used for OpenStack deploys.
atsMasterImage (optional)	Image used for OpenStack deploys.
atsAvailabilityZone (optional)	Name of logical partitioning in case of host aggregate.

Table 10-23 Parameters and Definitions for ATS Master (VNFM - V2)

Parameter	Description
vnfConfigurableProperties	Data required for VNF configuration.
ntpServerIp	IP of the NTP server.
imiNetwork	Network used for internal communication of DSR entities.

Note

The atsKeyName pair is created dynamically through VNFM. The same public key is put into all the ATS instances (master), and the private key is in the ATS master stack output. Use the same private key to log in to the ATS instance (master) by running the following command:

```
ssh -i <ats private key> <username>@<ats master Ip>
```

Example: `ssh -i atskey.pem cloud-user@10.75.189.120`

10.17 Non-ConfigDrive VNF Instantiation

By default config drive is enabled through VNFM.

While instantiating VNF through VNFM. It will use configuration drive feature of Openstack to fetch the data from openstack.

ConfigDrive feature must be enabled from Openstack and meta data must be disabled to use.

If any user does not want to use configDrive feature of Openstack, then while instantiating VNF through VNFM, the user must pass "configDrive": "false" through request body.

For example: In additional parameter

```
"additionalParams": {
  "ntpServerIp": "10.250.32.10",
  "xmiNetwork": {
    "name": "ext-net3",
    "subnet": [{
      "name": "ext-net3-subnet",
      "ipVersion": "IPv4"
    }]
  },
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [{
      "name": "imi-private-sub",
      "ipVersion": "IPv4"
    }]
  },
  "configDrive": "false"
```

}

10.18 Scale VNF to Level (Only Scale Out)

The N/B LCM scale_to_level Rest I/F helps in scaling existing VNF's.

Following are the available options while scaling using "scale to VNF level" N/B Interface:

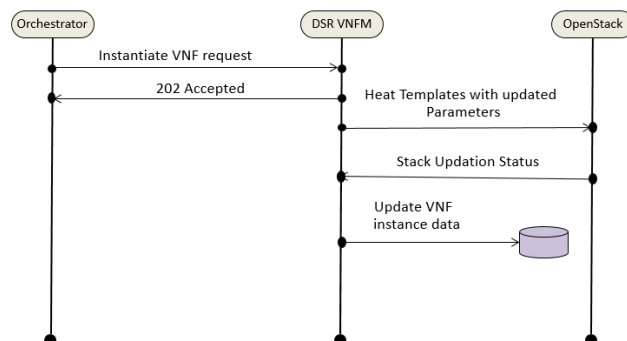
- Scale VNF to Level based on pre-defined sizes (using Instantiation level ID).
This option supports scaling of VNF from a lower instantiation level to higher, for example, Small to Medium.
- Scale VNF to Level with arbitrary sizes (using scaleInfo).
This option supports scaling of VNF to arbitrary sizes based on ScaleInfo. Scale VNF To Level using arbitrary size means increasing the existing VNFC count within the maximum allowed VNFC count. Note that maximum allowed VNFC count is the count from the existing VNFs flavorId with 'large' InstantiationLevelId. This count can be referred from respective VNF instantiation section. If the number of VMs to be scaled are greater than the default 'large' instantiationLevelId configurations, then change the configuration as described in [Configure Flavor and Instantiation Levels in VNFM](#). After that, trigger this REST request.

Note

- This feature is only supported for Scaling out C-level servers of Signaling Stack.
- The stack must have been instantiated prior to performing scale to level operation.
- Before Scaling the VNF to level, VnfInstance ID of the stack must be available.
- Scale to Level Request accepts either instantiationLevelId or scaleInfo.
- Cross deployment scaling is not supported by VNFM - if the user instantiated the VNF in fixed IP deployment model, then he must scale to level using FIXED IP deployment model only and vice versa.

The following image illustrates the VNF Scaling:

Figure 10-7 VNF Scaling



10.18.1 Scale VNF to Level using InstantiationLevelId for DSR Signaling

Note

SOAM Image version and scale Image (daScaleImage, ipfeScaleImage, stpScaleImage, sbrScaleImage, serviceMpScaleImage) version must be same.

Sample Request

Scaling VNF to Level Request for Dynamic IP model

Resource URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<VNF ID received from create/instantiate request>/scale_to_level`

Accept: `application/json`

Content-Type: `application/json`

```
{
  "instantiationLevelId": "medium",
  "additionalParams": {
    "daScaleImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeScaleImage": "DSR-9.0.2.0.0.vmdk",
    "stpScaleImage": "DSR-9.0.2.0.0.vmdk",
    "sbrScaleImage": "DSR-9.0.2.0.0.vmdk",
    "serviceMpScaleImage": "DSR-9.0.2.0.0.vmdk"
  }
}
```

Scaling VNF to Level Request for Fixed IP model

Resource URL: `https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/<VNF ID received from create/instantiate request>/scale_to_level`

Accept: `application/json`

Content-Type: `application/json`

Authorization : <Token generated after login>

```
{
  "instantiationLevelId": "medium",
  "additionalParams": {
    "daScaleImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeScaleImage": "DSR-9.0.2.0.0.vmdk",
    "stpScaleImage": "DSR-9.0.2.0.0.vmdk",
    "serviceMpScaleImage": "DSR-9.0.2.0.0.vmdk",
    "sbrScaleImage": "DSR-9.0.2.0.0.vmdk",
    "xmiNetwork": {
      "name": "ext-net",
      "subnet": [{
        "name": "ext-net-subnet",
        "ipVersion": "IPv4",
```

```

        "fixedIps":{
            "dampXmiIps":
["10.75.218.123", "10.75.218.21"],
            "ipfeXmiIps":
["10.75.218.3", "10.75.218.2"],
            "stpXmiIps":
["10.75.218.42", "10.75.218.143"],
            "serviceMpXmiIps": ["10.75.218.43",
"10.75.218.144"],
            "sbrXmiIps":
["10.75.218.23", "10.75.218.19"]
        }
    }],
    "imiNetwork": {
        "name": "imi-net",
        "subnet": [{
            "name": "imi-net-sub",
            "ipVersion": "IPv4",
            "fixedIps":{
                "dampImiIps":
["192.167.2.1", "192.167.2.2"],
                "ipfeImiIps":
["192.167.2.4", "192.167.2.3"],
                "stpImiIps":
["192.167.2.5", "192.167.2.6"],
                "serviceMpImiIps": ["192.167.2.9",
"192.167.2.10"],
                "sbrImiIps":
["192.167.2.7", "192.167.2.8"]
            }
        }]
    },
    "sbrNetwork":{
        "name": "ext-net2",
        "subnet": [{
            "name": "ext-net2-sub",
            "ipVersion": "IPv4",
            "fixedIps":{
                "sbrNetworkIps":
["10.75.219.23", "10.75.219.123"]
            }
        }]
    },
    "xsiNetwork": [{
        "name": "ext4-net2",
        "subnet": [{
            "name": "ext4-net2-sub",
            "ipVersion": "IPv4",
            "fixedIps":{
                "dampXsiIps":
["10.75.219.23", "10.75.219.12"],
                "ipfeXsiIps":
["10.75.219.1", "10.75.219.112"],
                "stpXsiIps":
["10.75.219.12", "10.75.219.23"]
            }
        }]
    }

```

```

    }
  }
}

```

① Note

The 202 response means that the request was accepted for processing. The VNF might take up to 6 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.

Sample Response

202 Accepted

Headers:

```

{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0  content-type:
  application/xml
}

```

Detailed explanation of XMI and XSI Network

① Note

- The instantiation level must be decided based on the number of VMs required.
- Only the IPs of the required VM are to be provided in the fixedIp parameter and they must be of the same network in that order as used during the instantiation process.

For Example:

"flavorId": "DIAMETER+STP", "instantiationLevelId": "medium" (scaling from small to medium) - This brings up 2 new DAMPs(DAMP02, DAMP03) , 2 new STP(STP 02, STP 03) servers.

The user needs to provide dampXmiIps(2), stpXmiIps(2), dampXsiIps(2), stpXsiIps(2)

① Note

If TSA is present while instantiation, then the scaled DAMP is automatically added to all the existing TSA.

The detailed explanation of XMI and XSI Network for the additional parameters is provided below:

For XMI Network

```

"xmiNetwork":{
  "name": "<Name of XMI network>",
  "subnet":[{
    "name": "<Name of Subnet of XMI Network>",
    "ipVersion": "",
    "fixedIps":{
      "dampXmiIps":[
        "<DAMP 02 XMI IP>",
        "<DAMP 03 XMI IP>"
      ],
      "stpXmiIps":[
        "<STP 02 XMI IP>",
        "<STP 03 XMI IP>"
      ]
    }
  ]
}

```

For IMI Network

```

"imiNetwork":{
  "name": "<Name of IMI Network>",
  "subnet":[{
    "name": "<Name of subnet of IMI Network>",
    "ipVersion": "",
    "fixedIps":{
      "dampImiIps":[
        "<DAMP 02 IMI IP>",
        "<DAMP 03 IMI IP>"
      ],
      "stpImiIps":[
        "<STP 02 IMI IP>",
        "<STP 03 IMI IP>"
      ]
    }
  ]
}

```

For XSI Network

```

"xsiNetwork":[
  {
    "name": "<Name of XSI-1 Network>"
    "subnet":[{
      "name": "<Name of Subnet of XSI-1 network>",
      "ipVersion": "",
      "fixedIps":{
        "dampXsiIps":[
          "<DAMP02 XSI 1 IP>",
          "<DAMP03 XSI 1 IP>"
        ],

```

```

    "stpXsiIps":[
        "<STP02 XSI 1 IP>",
        "<STP03 XSI 1 IP>"
    ]
}
]]
},
{
    "name": "<Name of XSI-2 Network>",
    "subnet": [{
        "name": "<Name of subnet of XSI-2 Network>",
        "ipVersion": "",
        "fixedIps":{
            "dampXsiIps":[
                "<DAMP02 XSI 2 IP>",
                "<DAMP03 XSI 2 IP>"
            ],
            "stpXsiIps":[
                "<STP02 XSI 2 IP>",
                "<STP03 XSI 2 IP>"
            ]
        }
    ]
}
]]
}
]

```

The following table describes the parameters used for sending request to VNFM:

Table 10-24 Scaling VNF to Level using InstantiationLevelId

Parameters	Definitions
instantiationLevelId	Identifier of the instantiation level of the deployment flavor to be scaled.
dampXmilps	List of DAMP external management IPs if new DAMP VMs are to be scaled.
ipfeXmilps	List of IPFE external management IPs if new IPFE VMs are to be scaled.
stpXmilps	List of vSTP external management IPs if new vSTP VMs are to be scaled.
sbrXmilps	List of SBR external management IPs if new SBR VMs are to be scaled.
sbrNetworkIps	List of SBR replication port IPs if new SBR VMs are to be scaled.
dampXsilps	List of DAMP signaling IPs if new DAMP VMs are to be scaled.
ipfeXsilps	List of IPFE signaling IPs if new DAMP VMs are to be scaled.
stpXsilps	List of STP signaling IPs if new DAMP VMs are to be scaled.
dampImilps	List of DAMP internal management Ips if new DAMP VMs are to be scaled.
ipfeImilps	List of IPFE internal management IPs if new IPFE VMs are to be scaled.

Table 10-24 (Cont.) Scaling VNF to Level using InstantiationLevelId

Parameters	Definitions
stpImlps	List of vSTP internal management IPs if new vSTP VMs are to be scaled.
sbrImlps	List of SBR internal management Ips if new SBR VMs are to be scaled.
subnet	List of subnet name and ipVersion used. It also contains fixed IPs if used.
daScaleImage (Optional)	DAMP Image used during scaling.
ipfeScaleImage (Optional)	IPFE Image used during scaling. If you are using the Multiqueue feature, you must follow the Prerequisite for the IPFE image.
stpScaleImage (Optional)	STP Image used during scaling.
serviceMpScaleImage (Optional)	Service Mp Image used during Scaling.
sbrScaleImage (Optional)	SBR Image used during scaling.

Note

During Scaling of SBRs, the newly spawned SBRs are not added to any Server Group. It needs to be added manually to the new Server Groups created by the user. One server group can have maximum two SBRs.

10.18.2 Scale VNF to Level using ScaleInfo

This option supports Scaling of VNF to arbitrary sizes based on **ScaleInfo**.

Scale VNF to Level using arbitrary size means increasing existing VNFC count within the max allowed VNFC count.

Max allowed VNFC count is the count from existing VNF's flavourId with Large InstantiationLevelId.

Note

Max allowed VNFC count can be referred from Instantiating the first signaling VNF section.

Sample Request:

Request URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/< VNF ID received from create/instantiate request>/scale_to_level`

Accept: `application/json`

Content-Type: `application/json`

Authorization: Token generated after login

Scaling VNF to Level Request for Dynamic IP deployment

```
{
  "scaleInfo": [{
    "aspectId": "DAMP",
    "scaleLevel": "3"
  }],
  "additionalParams": {
    "daScaleImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeScaleImage": "DSR-9.0.2.0.0.vmdk",
    "stpScaleImage": "DSR-9.0.2.0.0.vmdk",
    "serviceMpScaleImage": "DSR-9.0.2.0.0.vmdk",
    "sbrScaleImage": "DSR-9.0.2.0.0.vmdk",
  }
}
```

Scaling VNF to Level Request for Fixed IP deployment

```
{
  "scaleInfo": [
    {
      "aspectId": "DAMP",
      "scaleLevel": "4"
    },
    {
      "aspectId": "IPFE",
      "scaleLevel": "4"
    },
    {
      "aspectId": "STPMP",
      "scaleLevel": "4"
    },
    {
      "aspectId": "SERMP",
      "scaleLevel": "4"
    },
    {
      "aspectId": "SBR",
      "scaleLevel": "6"
    }
  ],
  "additionalParams": {
    "daScaleImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeScaleImage": "DSR-9.0.2.0.0.vmdk",
    "stpScaleImage": "DSR-9.0.0.2.0.vmdk",
    "serviceMpScaleImage": "DSR-9.0.2.0.0.vmdk",
    "sbrScaleImage": "DSR-9.0.2.0.0.vmdk",
  },
  "xmiNetwork": {
    "name": "ext-net",
    "subnet": [{
      "name": "ext-net-subnet",
      "ipVersion": "IPv4",
      "fixedIps": {
        "dampXmiIps": ["10.75.218.123", "10.75.218.21"],

```

```

"ipfeXmiIps": ["10.75.218.3", "10.75.218.2"],
"stpXmiIps": ["10.75.218.42", "10.75.218.143"],

"serviceMpXmiIps": ["10.75.218.42", "10.75.218.143"],
"sbrXmiIps": ["10.75.218.23", "10.75.218.19"]
}
}]
},
"imiNetwork": {
  "name": "imi-net",
  "subnet": [{
    "name": "imi-net-sub",
    "ipVersion": "IPv4",
    "fixedIps": {
      "dampImiIps": ["192.167.2.1", "192.167.2.2"],
      "ipfeImiIps": ["192.167.2.4", "192.167.2.3"],
      "stpImiIps": ["192.167.2.5", "192.167.2.6"],

      "serviceMpImiIps": ["192.167.2.5", "192.167.2.6"],
      "sbrImiIps": ["192.167.2.7", "192.167.2.8"]
    }
  }]
},
"sbrNetwork": {
  "name": "ext-net2",
  "subnet": [{
    "name": "ext-net2-sub",
    "ipVersion": "IPv4",
    "fixedIps": {
      "sbrNetworkIps": ["10.75.219.23", "10.75.219.123"]
    }
  }]
},
"xsiNetwork": [{
  "name": "ext4-net2",
  "subnet": [{
    "name": "ext4-net2-sub",
    "ipVersion": "IPv4",
    "fixedIps": {
      "dampXsiIps": ["10.75.219.23", "10.75.219.12"],
      "ipfeXsiIps": ["10.75.219.1", "10.75.219.112"],
      "stpXsiIps": ["10.75.219.12", "10.75.219.23"]
    }
  }]
}]
}
}

```

Note

The 202 response means that the request was accepted for processing. The VNF might take up to 6 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.

Sample Response

```

202 Accepted
Headers:
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0  content-type:
  application/xml
}

```

Note

- The aspect Id is decided based on the VM to be scaled, scale level is decided based on the number of VMs required.
- Only the IPs of the required VM must be provided in the fixedIp parameter and they must be of the same network in that order as used during the instantiation process.

For Example:

"aspectId":"DAMP", "scaleLevel":"4" (from scaleLevel 2 to scaleLevel 4) (scaling from small to medium) - This brings up 2 new DAMPs (DAMP02, DAMP03) servers.

The user needs to provide dampXmiIps(2), dampXsiIps(2), dampImiIps(2)

Note

If TSA is present while instantiation, then the scaled DAMP is automatically added to all the existing TSA.

Detailed explanation of XMI, IMI and XSI Network

The detailed explanation of XMI , IMI and XSI Network for the additional parameters is provided below:

For XMI Network

```

"xmiNetwork":{
  "name": "<Name of XMI network>",
  "subnet": [{
    "name": "<Name of Subnet of XMI network>",
    "ipVersion": "",
    "fixedIps":{
      "dampXmiIps":[
        "<DAMP 02 XMI IP>",
        "<DAMP 03 XMI IP>"
      ]
    }
  ]
}

```

For IMI Network

```

"imiNetwork":{
  "name": "<Name of IMI network>",
  "subnet": [{
    "name": "<Name of subnet of IMI network>",
    "ipVersion": "",
    "fixedIps":{
      "dampImiIps":[
        "<DAMP 02 IMI IP>",
        "<DAMP 03 IMI IP>"
      ]
    }
  ]
}

```

For XSI Network

```

"xsiNetwork":[
  {
    "name": "<Name of XSI-1 Network>",
    "subnet":[{
      "name": "<Name of subnet of XSI-1 Network>",
      "ipVersion": "",
      "fixedIps":{
        "dampXsiIps":[
          "<DAMP02 XSI 1 IP>",
          "<DAMP03 XSI 1 IP>"
        ]
      }
    }
  },
  {
    "name": "<Name of XSI-2 Network>",
    "subnet":[{
      "name": "<Name of Subnet of XSI-2 Network>",
      "ipVersion": "",
      "fixedIps":{
        "dampXsiIps":[
          "<DAMP02 XSI 2 IP>",
          "<DAMP03 XSI 2 IP>"
        ]
      }
    }
  ]
}

```

The following table describes the parameters used for sending request to VNFM:

Table 10-25 Parameters and Definitions for Scaling VNF to Level using ScaleInfo

Parameters	Definitions
scaleInfo	aspectId : VnfType scaleLevel : Target scale level to which the VNF is to be scaled
dampXmilps	List of DAMP external management ips (if new DAMP VMs are to be scaled)
ipfeXmilps	List of IPFE external management ips (if new IPFE VMs are to be scaled)
stpXmilps	List of vSTP external management ips (if new vSTP VMs are to be scaled)
sbrXmilps	List of SBR external management ips (if new SBR VMs are to be scaled)
sbrNetworkIps	List of SBR replication port ips (if new SBR VMs are to be scaled)
dampXsilps	List of DAMP signaling ips (if new DAMP VMs are to be scaled)
ipfeXsilps	List of IPFE signaling ips (if new DAMP VMs are to be scaled)
stpXsilps	List of STP signaling ips (if new DAMP VMs are to be scaled)
dampImilps	List of DAMP internal management ips (if new DAMP VMs are to be scaled)
ipfeImilps	List of IPFE internal management ips (if new IPFE VMs are to be scaled)
stpImilps	List of vSTP internal management ips (if new vSTP VMs are to be scaled)
sbrImilps	List of SBR internal management ips (if new SBR VMs are to be scaled)
daScaleImage (Optional)	DAMP Image used during scaling
ipfeScaleImage (Optional)	IPFE Image used during scaling If you are using the Multiqueue feature, you must follow the Prerequisite for the IPFE image.
stpScaleImage (Optional)	STP Image used during scaling
serviceMpScaleImage (Optional)	Service Mp Image used during Scaling
sbrScaleImage (Optional)	SBR Image used during scaling

10.18.3 IPFE and Service Mp Scale from Sizing 0

This feature supports scale from sizing 0 for IPFE and Service Mp VMs.

- For IPFE, we can scale from size 0 by using "instantiationLevelId" and ScaleInfo (Arbitrary Size).
- For Service Mp, we can scale from size 0 by using only ScaleInfo (Arbitrary Size).

Note

Service Mp Instantiation and scale introduced from DSR version 8.5.0.1.0-91.14.0.

10.19 Operate VNF

Operate VNF enables requesting to change the operational state of a VNF instance, including starting and stopping the VNF instance. The VNF state must be in the INSTANTIATED state.

A VNF instance can be in the following states:

- **STARTED:** Indicates that the VNF instance is up and running.
- **STOPPED:** Indicates that the VNF instance has been stopped. A VNF instance is stopped if all its VNFC instances are also stopped.

In addition to the VNF level Operate, it supports Operate VNF Stop and Start at the VNFC (VM) level. The Operate API can take the VM Hostname in the request message when the call is for the operation of the specific VNFCs.

Operate VNF supports both GRACEFUL and FORCEFUL stopType. If StopType is GRACEFUL, gracefulStopTimeout (in seconds) must be passed.

StopType is not required if chageStateTo is STARTED.

A VNF instance is STARTED/STOPPED if all of its VNFC instances are also in the STARTED/STOPPED state. "vnfState" as STARTED/STOPPED/PARTIAL are implemented. "vnfState" remains in the "PARTIAL" state if VNFC states are combination of "STARTED" and "STOPPED" states. Instance file structure for "additionalInfo" is updated to specific format to support Operate API call after Instantiate, Scale To Level and Discover of VNF. There is a retry mechanism for all VNFCs to complete the operation. If VNFC fails to operate, LCM status returns as FAILED updating the instance file.

Note

- Operate VNF is a maintenance API, which is similar to shutdown VNF. The orchestrator decides when to call Operate after completing the impact assessment.
- If the traffic impact is high, isolate the VM, and then run the shutdown. For example, remove the DA_MP from the TSA group manually or remove the DP from the DP server group, and then run shut off Instance by performing operate VNF API call to change the VFNC state to the STOPPED state.

GRACEFUL StopType

- Operate VNF supports GRACEFUL stopType.
- **gracefulStopTimeout:** The time interval (in seconds) to wait for the VNF to be taken out of service during graceful stop before stopping the VNF.

Note

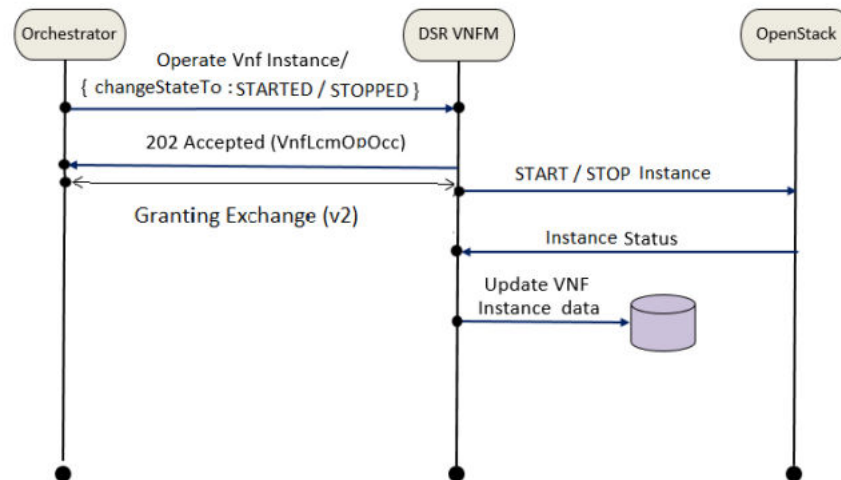
You must manually perform the following procedure before doing GRACEFUL STOP for a particular VNF.

1. Open corresponding Active NOAM GUI.

2. On the **Status & Manage** tab, under HA - edit the **Max Allowed HA** Role of required instances of the vnf as OOS.
3. Proceed with GRACEFUL stop operate vnf operation from VNFM.
4. After change State to STARTED, you can manually edit the **Max Allowed HA** Role of required instances of the vnf as **Active** state.

The following image displays the VNF Operate:

Figure 10-8 Operate vnf



Sample Request for Operate VNF

URL: `https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/< VNF ID received from create/instantiate request>/operate`

```

Accept: application/json
Content-Type: application/json
Authorization : <Token generated after login>
{
  "changeStateTo": "STARTED",
  "additionalParams": {
    "vnfcNames": ["Opt-DN001"]
  }
}

```

Note

The 202 response indicates that the request is accepted for processing. Use the Query LCM operation for status.

Operate VNF Response

202 Accepted
Headers:

```
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6

  date: Tue, 29 Jan 2019 10:39:24 GMT

  content-length: 0  content-type:
    application/xml
}
```

Sample Request for Operate VNF GRACEFUL STOP

RL: https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/< VNF
ID received from create/instantiate request>/operate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```
{
  "changeStateTo": "STOPPED",
  "stopType": "FORCEFUL",
  "additionalParams":

  {
    "vnfcNames": ["dsNO-DN001"]
  }
}
```

Note

The 202 response means that the request was accepted for processing. Use Query LCM operation for status.

Operate VNF Response

202 Accepted

Headers:

```
{

  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
```

```

    date: Tue, 29 Jan 2019 10:39:24 GMT
    content-length: 0 content-type:
    application/xml
}

```

Sample Request for Operate VNF FORCEFUL STOP

Operate VNF

```

URL: https://<>:8443/vnfm/<>/vnf_instances/<
      VNF ID received from create /instantiate request>/operate
Accept: application/json
Content-Type: application/json
Authorization: <Token generated login>
{
  "changeStateTo": "STOPPED",
  "stopType": "FORCEFUL",
  "additionalParams":
  { "vnfcNames": ["dsNO-DN001"]
  }
}

```

Note

The 202 response means that the request was accepted for processing. Use Query LCM operation for status.

Operate VNF Response

```

202 Accepted
Headers:
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cda03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0 content-type:
  application/xml
}

```

10.19.1 Operate VNF Parameter Descriptions

The following table describes Operate VNF parameters:

Table 10-26 Operate VNF Parameter Descriptions

Parameter	Descriptions
changeStateTo	Changes the operational state to STARTED or STOPPED.

Table 10-26 (Cont.) Operate VNF Parameter Descriptions

Parameter	Descriptions
stopType (Optional)	GRACEFUL / FORCEFUL. stopType is required if changeStateTo is STOPPED .
gracefulStopTimeout (Optional)	gracefulStopTimeout is required if changeStateTo is STOPPED and stopType is GRACEFUL . The time interval (in seconds) to wait for the VNF to be taken out of service during graceful stop before stopping the VNF.
vnfcNames (Optional)	Changes the operational state of a particular VNFCs to STARTED or STOPPED.

11

VNF Instantiation across Multi Cloud / Multi Tenant / Multi Network

VNFM supports multi-cloud, multi-tenant, and multi network deployment for DSR and SDS VNF.

List of VNF deployment of multi cloud/tenant:

Table 11-1 Multi cloud/tenant deployment

Tenant-1/Cloud-1	Tenant-2/Cloud-2
DSR-NOAM	DSR-Signaling
DSR-NOAM	DSR-DR-NOAM
SDS-NOAM	SDS-Signaling
SDS-NOAM	SDS-DR-NOAM

List of VNF deployment of multi network:

Table 11-2 Multi Network Deployment

Network-1	Network-2
DSR-NOAM	DSR-DR-NOAM
SDS-NOAM	SDS-DR-NOAM

Note

- While deploying DSR Signaling/DSR DR VNF, vnfInstanceId of DSR Noam should be passed in additional params.
- While deploying SDS Signaling/SDS DR VNF, vnfInstanceId of SDS Noam should be passed in additional params.
- The "vnfInstanceId" is the mandatory parameter while multi-cloud/tenant VNF deployment only incase of passing OpenStack resource IDs for the XMI IPs from both NOAM VMs.

Sample Request

Sample Request Body of additional parameter changes for DSR Signaling VNF in case of multi tenant/cloud

```
"additionalParams": {  
    "xmiNetwork": {  
        "name": "ext-net3",  
        "subnet": [{  
            "name": "ext-
```

```

net3-subnet",
                                "ipVersion":
                                "IPv4"
                                ]]
                                },
                                "imiNetwork": {
                                    "name": "imi-private",
                                    "subnet": [{
                                        "name": "imi-
private-sub",
                                        "ipVersion":
                                        "IPv4"
                                        ]]
                                },
                                "xsiNetwork": [{
                                    "name": "ext-net2",
                                    "subnet": [{
                                        "name": "ext-
net2-sub",
                                        "ipVersion":
                                        "IPv4"
                                        ]]
                                },
                                {
                                    "name": "ext-net5",
                                    "subnet": [{
                                        "name": "ext-
net5-sub",
                                        "ipVersion":
                                        "IPv4"
                                        ]]
                                }
                                ],
                                "ntpServerIp": "10.250.32.10",
                                "primaryNoamVmName": "NOAM00-32cd6138",
                                "noamSgName":
                                "dsrNetwork0am_NOAM_32cd6138_SG",
                                "vnfInstanceId":
                                "dsrNetwork0am-4e99a1cd-77b7-478b-9b28-32cd6138"
                                }

```

12

Heal VNF

Using the Heal VNF operation, you can request for healing a VNFC instance. Auto-healing is not supported by VNFM. Only Orchestrator-triggered healing is supported.

Heal VNF accepts a single VNFC name per healing request. Only one server is healed at a certain point of time per each (DSR/SDS) NOAM topology to avoid complete state down. When there are concurrent healing requests for multiple VNFCs from the same system topology, the requests are arranged in a queue and are processed in the same sequence as they are positioned in the queue.

Heal VNF Resource only supports the following VNFs with DSR/SDS version 8.5.0.1.0 or later:

- DSR Network OAM
- DSR DR Network OAM
- DSR Signaling
- SDS Network OAM
- SDS DR Network OAM
- SDS Signaling

Heal VNF supports the following DSR/SDS alarms:

Table 12-1 DSR/SDS Alarms

Alarm ID	Alarm Name	Healing Action
32306	Server RAM Shortage Error	Server Reboot
32307	Server Swap Space Shortage Error	Server Reboot
31101	DB Replication To Slave Failure	Server Reboot
31283	Lost Communication with server	Server Reboot
31114	DB Replication over SOAP has failed	Server Reboot
31233	HA Path Down	Server Reboot
31107	DB Merge From Child Failure	Server Reboot

Note

- Server Reboot and Application Restart are the only supported healing tasks.
- Heal Operation does not support recreation of any VNF/VNFC.
- When a VNFM state is changed as a part of Geo Redundancy, then all the healing operations waiting in the queue are discarded.

Sample Request for Heal VNF Request

URL: https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request>/heal

```
Accept: application/json
{
  "cause": "merging alarms",
  "additionalParams": {
    "vnfcName": "heal1-SN000",
    "noamVip": "10.75.189.34",
    "username": "root",
    "password": "xxxx"
  }
}
```

Note

In the heal request body username and password are optional.

Sample Response for Heal VNF Request

```
Response Code : 202
{
  location: https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0 content-type:
  application/xml
}
```

12.1 Query Heal VNF Requests

Query Heal VNF requests provide the list of all the Heal VNF requests present in the Queue corresponding to a single (DSR/SDS) NOAM VIP. The URL accepts a (DSR/SDS) NOAM VIP.

Sample Request for Query Heal VNF Requests

URL:

https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vips/<<VIP>>/heal

Sample Response for Query Heal VNF Requests

```
Response Code : 200
{
  "10.75.225.207": [
    {
      "hostName": "DisNo2-DS001",
      "level": "B",
      "role": "Soam",

```

```

        "vnfInstanceId": "dsrSignaling-c0732735-cbd3-4a81-8d05-4804d631ab37",
        "status": "PROCESSING"
    },
    {
        "hostName": "DisNo-DN001",
        "level": "A",
        "role": "Noam",
        "vnfInstanceId": "dsrNetwork0am-c0732795-cbd3-4a81-8d05-4804d638aaa3",
        "status": "IN_QUEUE"
    }
]
}

```

12.1.1 Heal VNF Parameter Descriptions

The following table describes Heal VNF parameters:

Table 12-2 Heal VNF Parameter Descriptions

Parameter	Descriptions
cause	The cause for issuing this healing request.
vnfcName	The name of the instance/VNFC that needs to be healed.
noamVip	DSR/SDS NOAM VIP of the topology to which the given VNFC belongs to.
username (optional)	The SSH login username of the NOAM. admusr is the default value.
password (optional)	The SSH login password of the NOAM. The default password for admusr is used as the default value.

13

Discover Stack

It is an N/B LCM Discover Rest Interface. The orchestrator can use this information to scale out the stack. Before discovering the stack, ensure that you have the following information available:

- The Stack ID of the previously created stack.
- Details about the OpenStack instance where the stack will be discovered:
 - OpenStack Controller URI
 - User Domain Name
 - Project Domain Id
 - Username
 - Password
 - Project name
- The Interface discovers the stack and performs the following operations:
 - Downloads the parameter file of the discovered stack.
 - Creates the Instance file of the discovered stack.
 - These files are saved in the /var/vnfm/instances/<autoDiscovery InstanceId>/ directory.
 - Generates a VNFD file in the same location for V2.

Sample Request for Discover Interface

Request URL : POST:

```
https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/discover/<<discover  
stack id>>
```

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

Request Body:

```
{  
  "vimConnectionInfo": {  
    "MvlCloud": {  
      "vimId": "vimId",  
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",  
      "interfaceInfo": {  
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"  
      },  
      "accessInfo": {  
        "project": "*****",  
        "region": "RegionOne",  
        "username": "*****",  
        "password": "*****",  
      }  
    }  
  }  
}
```

```
        "userDomain": "Default",  
        "projectDomain": "default"  
    }  
},  
"vnfSoftwareVersion": "DSR-9.0.2.2.0_99.16.0"  
}
```

Sample Response for Discover Interface

Response Code: 200

```
{  
  "vnfInstanceId": "dsrNetwork0am-945cffa107c235bb-43d87678-756b-4f8e-a59c-  
d9b7d4dd95a1",  
  "discoverStackId": "7d861391-0ed2-4d0b-9f01-e84e186e9244"  
}
```

Note

- The Discover VNF Stack feature supports stacks created by the VNFM GUI and recovers from a double failure of the active VNFM and its persistent volume.
- The `vnfSoftwareVersion` parameter is optional. If provided with a DSR version, the same version is displayed in the Query VNF API response.
- This feature ensures VNFM backward compatibility. Upgraded DSR versions are supported with newly installed compatible VNFM versions.

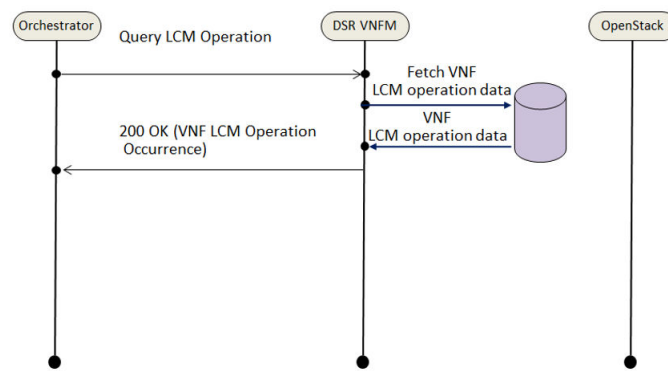
14

Query LCM Operation

This resource represents VNF lifecycle management operation occurrences. This resource can be used to query status information about multiple VNF lifecycle management operation occurrences.

The following image illustrates the sequence for querying/reading information about a VNF LCM Operation.

Figure 14-1 VNF LCM Operation



Query LCM Operation, using the following two ways:

- Query individual LCM Operation
- Query All LCM Operation

14.1 Query Individual LCM Operation

If the NFVO intends to read information about a particular LCM Operation, it sends a GET request to the "Individual LCM operation" resource, addressed by the appropriate VNF LCM Operation occurrence identifier (vnfLcmOp0ccId) in its resource URI.

The VNFM returns a **200 OK** response to the NFVO, and includes specific data structure of type "VnfLcmOp0cc" related to the VNF LCM Operation occurrence identifier (vnfLcmOp0ccId) in the payload body.

Sample Request

Query individual LCM Operation

URL: GET: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs /<<{vnfLcmOp0ccId}>>`

Sample Response

URL: GET: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/<<{vnfLcmOp0ccId}>>`

```

Accept: application/json
Content-Type: application/json
Authorization: Token generated after login
{
  "id": "lcmOp-00301ea4-a7b2-4334-8b93-190377700ab0",
  "operationState": "COMPLETED",
  "stateEnteredTime": "2019/02/08 07:33:00 UTC",
  "startTime": "2019/02/08 07:31:19 UTC",
  "vnfInstanceId": "dsrNetwork0am-cf67bff6-e9c9-4213-b6fa-b5337c3d30b6",
  "operation": "TERMINATE",
  "operationParams": {
    "terminationType": "FORCEFUL",
    "gracefulTerminationTimeout": null,
    "additionalParams": null
  },
  "links": {
    "self": {
      "href": "https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs/lcmOp-00301ea4-a7b2-4334-8b93-190377700ab0"
    },
    "vnfInstance": {
      "href": "https://localhost:8443/vnflcm/<<vnfm_version>>/vnf_instances/dsrNetwork0am-cf67bff6-e9c9-4213-b6fa-b5337c3d30b6"
    }
  },
  "isCancelPending": false,
  "isAutomaticInvocation": false
}

```

Sample Response for Individual LCM Operation:

Example of the LCM successful scenario:

Response Code : 200 URL: GET: https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_lcm_op_occs/<<{vnfLcmOp0ccId}>>

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

Response Body:

```

{
  "id" : "lcmOp-4d08edd3-4e62-43eb-b9f8-7fe0cd6bbb2d",
  "operationState" : "COMPLETED",
  "stateEnteredTime" : "2023/03/07 07:22:59 UTC",
  "startTime" : "2023/03/07 07:22:07 UTC",
  "vnfInstanceId" : "2d7e5480-5e68-4a3e-b1a0-c6931afc895f",
  "operation" : "TERMINATE",
  "grantId" : "15e2decf-7e75-4091-8b5b-2b087f404bda",
  "operationParams" : {
    "terminationType" : "FORCEFUL",
    "gracefulTerminationTimeout" : null,
    "additionalParams" : null
  },
}

```

```

"resourceChanges" : {
  "affectedVnfcs" : [ {
    "id" : "v2S0-DS000",
    "vduId" : "soam",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "ce851362-4f7e-4d88-bbce-a06dd16ada5f"
    }
  }, {
    "id" : "v2S0-DS001",
    "vduId" : "soam",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "53579d61-c45d-4d45-81b5-6adfee96f477"
    }
  }, {
    "id" : "v2S0-DMP00",
    "vduId" : "damp",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "cad6a1c3-9027-4997-9c83-35813343cac3"
    }
  }, {
    "id" : "v2S0-DMP01",
    "vduId" : "damp",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "8c15c04e-7ba3-4dce-9a2a-ce20a893420b"
    }
  }, {
    "id" : "v2S0-DMP02",
    "vduId" : "damp",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "4ff8818e-535b-495f-931b-4006beac523c"
    }
  }, {
    "id" : "v2S0-DIP00",
    "vduId" : "ipfe",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "b01276ce-47ca-436a-bcde-2f7febdfdf273"
    }
  }, {
    "id" : "v2S0-DIP01",
    "vduId" : "ipfe",
    "changeType" : "REMOVED",
    "computeResource" : {
      "vimConnectionId" : "MvlCloud",
      "resourceId" : "044caa79-f09b-4f0c-b315-97edfd0465db"
    }
  }
]

```

```

    }
  } ]
},
"links" : {
  "self" : {
    "href" : "https://localhost:8443/vnflcm/v2/vnf_lcm_op_occs/
lcmOp-4d08edd3-4e62-43eb-b9f8-7fe0cd6bbb2d"
  },
  "vnfInstance" : {
    "href" : "https://localhost:8443/vnflcm/v2/vnf_instances/
2d7e5480-5e68-4a3e-b1a0-c6931afc895f"
  }
},
"isAutomaticInvocation" : false,
"isCancelPending" : false
}

```

Example of the FAILED LCM scenario:

Response Code: 200 URL: GET: https://<<VNFM HOST IP>>:8443/vnfm/
<<vnfm_version>>/vnf_lcm_op_occs/<<{vnfLcmOpOccId}>>

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

Response Body:

```

{
  "id": "lcmOp-536b138e-192d-4659-8457-b9306d4ced88",
  "operationState": "FAILED",
  "stateEnteredTime": "2021/07/20 07:12:00 GMT",
  "startTime": "2021/07/20 06:27:45 GMT",
  "vnfInstanceId": "dsrNetworkOam-cb8802a0-6358-4e78-805d-392f8bc9848b",
  "operation": "INSTANTIATE",
  "operationParams": {
    "flavourId": "DSR NOAM",
    "instantiationLevelId": "ha",
    "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [],
    "vimConnectionInfo": {
      "MvlCloud": {
        "vimId": "vimId",
        "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
        "interfaceInfo": {
          "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        }
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  }
}

```

```

    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "ntpServerIp": "10.250.32.10",
    "xmiNetwork": {
      "name": "ext-net5-dual",
      "subnet": [
        {
          "ipVersion": "IPv4",
          "name": "ext-net5-ipv4"
        }
      ]
    },
    "imiNetwork": {
      "name": "imi-private",
      "subnet": [
        {
          "ipVersion": "IPv4",
          "name": "imi-private-sub"
        }
      ]
    }
  }
},
"error": {
  "title": "DsrVnfmGenException",
  "status": 500,
  "detail": "There is cloud-init failure."
},
"resourceChanges": {
  "affectedVnfc": [
    {
      "id": "FT--8716b8dc-DN000",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "ad5b378e-40b7-4a83-a31d-ed7b9ddf2501"
      }
    },
    {
      "id": "FT--8716b8dc-DN001",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "dfae8cfd-94c4-4e1a-9322-6fc3e583243b"
      }
    }
  ]
},
"links": {
  "self": {
    "href": "https://[2606:b400:605:b813::b]:8443/vnflcm/v1/vnf_lcm_op_occs/
lcmOp-536b138e-192d-4659-8457-b9306d4ced88"
  },
  "vnfInstance": {

```

```

        "href": "https://[2606:b400:605:b813::b]:8443/vnflcm/v1/vnf_instances/
dsrNetwork0am-cb8802a0-6358-4e78-805d-392f8bc9848b"
    },
    "isAutomaticInvocation": false,
    "isCancelPending": false
}

```

Example of LCM operation after MODIFY VNF:

Response Code : 200 URL : GET: https://<<VNFM HOST IP>>:8443/vnfm/
<<vnfm_version>>/vnf_lcm_op_occs/<<{vnfLcmOpOccId}>>

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

Response Body:

```

{
  "id": "lcmOp-788ddde7-d06b-46a1-abf5-ddb782c23889",
  "operationState": "COMPLETED",
  "stateEnteredTime": "2023/03/06 09:25:03 UTC",
  "startTime": "2023/03/06 09:25:02 UTC",
  "vnfInstanceId": "2d7e5480-5e68-4a3e-b1a0-c6931afc895f",
  "operation": "MODIFY_INFO",
  "operationParams": {
    "vnfInstanceDescription": "123456",
    "vimConnectionInfo": null
  },
  "changedInfo": {
    "vnfInstanceDescription": "123456"
  },
  "links": {
    "self": {
      "href": "https://localhost:8443/vnflcm/v2/vnf_lcm_op_occs/
lcmOp-788ddde7-d06b-46a1-abf5-ddb782c23889"
    },
    "vnfInstance": {
      "href": "https://localhost:8443/vnflcm/v2/vnf_instances/
2d7e5480-5e68-4a3e-b1a0-c6931afc895f"
    }
  },
  "isAutomaticInvocation": false,
  "isCancelPending": false
}, {
  "id": "lcmOp-df76590e-5abd-4a85-a32f-db136600eae4",
  "operationState": "STARTING",
  "stateEnteredTime": "2023/03/06 09:27:50 UTC",
  "startTime": "2023/03/06 09:27:50 UTC",
  "vnfInstanceId": "2d7e5480-5e68-4a3e-b1a0-c6931afc895f",
  "operation": "MODIFY_INFO",
  "operationParams": {
    "vnfInstanceDescription": "test",
    "vimConnectionInfo": {

```

```

    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    },
    "links": {
      "self": {
        "href": "https://localhost:8443/vnflcm/v2/vnf_lcm_op_occs/
lcmOp-df76590e-5abd-4a85-a32f-db136600eae4"
      },
      "vnfInstance": {
        "href": "https://localhost:8443/vnflcm/v2/vnf_instances/
2d7e5480-5e68-4a3e-b1a0-c6931afc895f"
      }
    },
    "isAutomaticInvocation": false,
    "isCancelPending": false
  }
}

```

14.2 Query All LCM Operation

If the NFVO intends to query all LCM Operation, it sends a GET request to the **LCM operation** resource.

The VNFM returns a **200 OK** response to the NFVO, and includes zero or more data structures of type "VnfLcmOp0cc" in the payload body.

Sample Request

Query All LCM Operation

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs

Sample Response

```

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/
vnf_lcm_op_occs
Accept: application/json
Content-Type: application/json
Response Body for No VNF Instances
[]

```

```

Response Body for Query All LCM Operation
[
{

```

```

    "id": "lcmOp-ec72c7b4-7cea-4201-a0ab-5c0cec66cfa6",
    "operationState": "STARTING",
    "stateEnteredTime": "2019/01/16 05:53:31 UTC",
    "startTime": "2019/01/16 05:53:31 UTC",
    "vnfInstanceId": "dsrNetwork0am-dfc4dcd2-2752-48b4-875d-6cf703ba4134",
    "operation": "INSTANTIATE",
    "operationParams": {
      "flavourId": "DSR NOAM",
      "instantiationLevelId": "small11",
      "extVirtualLinks": "extVirtualLinks",
      "extManagedVirtualLinks": [],
    },
    "vimConnectionInfo": {
      "MvlCloud": {
        "vimId": "vimId",
        "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
        "interfaceInfo": {
          "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        },
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      },
    },
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "ntpServerIp": "10.250.32.10",
    "xmiNetwork": {
      "name": "ext-net7",
      "ipVersion": "IPv4",
      "xmiSubnetName": "ext-net7-subnet"
    },
  },
  "links": {
    "self": {
      "href": "https://localhost:8443/vnflcm/v1/vnf_lcm_op_occs/lcmOp-ec72c7b4-7cea-4201-a0ab-5c0cec66cfa6"
    },
    "vnfInstance": {
      "href": "https://localhost:8443/vnflcm/v1/vnf_instances/dsrNetwork0am-dfc4dcd2-2752-48b4-875d-6cf703ba4134"
    },
  },
  "isAutomaticInvocation": false,
  "isCancelPending": false
},
{
  "id": "lcmOp-00574fa7-8c4a-45ac-b7a8-816bfaf70985",
  "operationState": "STARTING",
  "stateEnteredTime": "2019/01/16 06:05:32 UTC",
  "startTime": "2019/01/16 06:05:32 UTC",

```

```

"vnfInstanceId": "dsrSignaling-08db63da-6cac-495f-8480-baf368d21cf7",
"operation": "INSTANTIATE",
"operationParams": {
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [
    {
      "id": "id1",
      "resourceId": "31ae9c8b-519e-4316-9a24-45c619646d69"
    },
    {
      "id": "id2",
      "resourceId": "aa9d142d-89d4-40e7-a701-559a993aa5ea"
    }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net7",
      "ipVersion": "IPv4",
      "xmiSubnetName": "ext-net7-subnet"
    },
    "xsiNetwork": {
      "name": "ext-net7",
      "ipVersion": "IPv4",
      "xsiSubnetName": "ext-net7-subnet"
    },
    "ntpServerIp": "10.250.32.10",
    "primaryNoamVmName": "NOAM00-03ba4134",
    "noamSgName": "dsrNetwork0am_NOAM_03ba4134_SG"
  },
  "links": {
    "self": {
      "href": "https://localhost:8443/vnflcm/v1/vnf_lcm_op_occs/
lcmOp-00574fa7-8c4a-45ac-b7a8-816bfaf70985"
    },
    "vnfInstance": {

```

```

        "href": "https://localhost:8443/vnflcm/v1/vnf_instances/
dsrSignaling-08db63da-6cac-495f-8480-baf368d21cf7"
    }
  },
  "isAutomaticInvocation": false,
  "isCancelPending": false
}
]

```

14.3 Query All LCM Operation Using Filter

VNFM supports query all LCM with filter option, where the user need to pass filter data in a specific format. It support operations such as eq, in, nin, cont, ncont, gte, and lte.

The VNFM returns a **200 OK** response and includes zero or more data structures of type "VnfLcmOp0cc" in the payload body.

Format: (operation, field, value)

Example: (eq, operationState, COMPLETED)

Sample Request for Query All LCM Operation with Filter:

Querying All LCM Operation: URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_lcm_op_occs.

Authorization : <Token generated after login>

Filter: (gte,stateEnteredTime,2023/07/08 10:16:00 UTC)

Sample Response:

```

[
  {
    "id": "lcmOp-9cd65a60-6731-4e1e-8ec1-8db670d910ea",
    "operationState": "FAILED",
    "stateEnteredTime": "2023/07/08 12:09:51 GMT",
    "startTime": "2023/07/08 10:16:52 GMT",
    "vnfInstanceId": "dsrSignaling_DRA-7ecac352-112a-41e1-9d15-1f0c5b789b6f",
    "operation": "INSTANTIATE",
    "operationParams": {
      "flavourId": "DIAMETER+STP",
      "instantiationLevelId": "small",
      "extVirtualLinks": "extVirtualLinks",
      "extManagedVirtualLinks": [
        {
          "resourceId": "10.75.189.241"
        },
        {
          "resourceId": "10.75.189.196"
        }
      ]
    },
    "vimConnectionInfo": {
      "MvlCloud": {
        "vimId": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {

```

```

        "endpoint": "https://mv1-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
        "username": "dsrvnfm",
        "password": "YXV0b21hdGlvbg==",
        "userDomain": "default",
        "projectDomain": "default",
        "project": "VNFM_FT1",
        "region": "VNFM_FT1"
    },
    "extra": {}
}
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
    "xmiNetwork": {
        "name": "ext-net",
        "vipSubnetName": "ext-net-ipv6-subnet",
        "subnet": [
            {
                "name": "ext-net-subnet",
                "ipVersion": "IPv4"
            },
            {
                "name": "ext-net-ipv6-subnet",
                "ipVersion": "IPv6"
            }
        ]
    },
    "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [
            {
                "name": "test11",
                "ipVersion": "IPv4"
            },
            {
                "name": "test6",
                "ipVersion": "IPv6"
            }
        ]
    },
    "soamAffinityPolicy": "soft-anti-affinity",
    "daAffinityPolicy": "soft-anti-affinity",
    "ipfeAffinityPolicy": "soft-anti-affinity",
    "stpAffinityPolicy": "soft-anti-affinity",
    "sbrAffinityPolicy": "soft-anti-affinity",
    "udrAffinityPolicy": "soft-anti-affinity",
    "xsiNetwork": [
        {
            "name": "xsiNetworkDual",
            "subnet": [
                {
                    "name": "xsiIPv4",
                    "ipVersion": "IPv4"
                }
            ]
        }
    ]
}

```

```

        },
        {
            "name": "xsiIPv6",
            "ipVersion": "IPv6"
        }
    ]
},
{
    "name": "xsiNetworkDual2",
    "subnet": [
        {
            "name": "xsiNetworkDual2-IPv4",
            "ipVersion": "IPv4"
        },
        {
            "name": "xsiNetworkDual2-IPv6",
            "ipVersion": "IPv6"
        }
    ]
}
],
"primaryNoamVmName": "FT--8bedbd47-DN000",
"noamSgName": "FT__8bedbd47_DN0SG"
}
},
"error": {
    "title": "DsrVnfmGenException",
    "status": 500,
    "detail": "There is cloud-init failure."
},
"resourceChanges": {
    "affectedVnfcS": [
        {
            "id": "FT--37bfc2c-DS000",
            "vduId": "soam",
            "changeType": "ADDED",
            "computeResource": {
                "vimConnectionId": "vimid",
                "resourceId": "50d32f55-c1e7-46de-985c-5669269a6592"
            }
        },
        {
            "id": "FT--37bfc2c-DS001",
            "vduId": "soam",
            "changeType": "ADDED",
            "computeResource": {
                "vimConnectionId": "vimid",
                "resourceId": "8c51db6a-2fb1-4bce-b457-d5e057e3ed8e"
            }
        },
        {
            "id": "FT--37bfc2c-DMP00",
            "vduId": "damp",
            "changeType": "ADDED",
            "computeResource": {
                "vimConnectionId": "vimid",

```

```

        "resourceId": "5986a9e9-7b3c-42b8-84b8-423fe5f9d2d3"
    },
    {
        "id": "FT--37bfc2c-DMP01",
        "vduId": "damp",
        "changeType": "ADDED",
        "computeResource": {
            "vimConnectionId": "vimid",
            "resourceId": "0f822a71-0454-45ad-8ef3-5a40063cea5c"
        }
    },
    {
        "id": "FT--37bfc2c-DIP00",
        "vduId": "ipfe",
        "changeType": "ADDED",
        "computeResource": {
            "vimConnectionId": "vimid",
            "resourceId": "916d3f0f-ddc3-408e-8056-b1e879867940"
        }
    },
    {
        "id": "FT--37bfc2c-DIP01",
        "vduId": "ipfe",
        "changeType": "ADDED",
        "computeResource": {
            "vimConnectionId": "vimid",
            "resourceId": "ff1fe735-f63c-45d6-a4f6-4fbaf22eeb58"
        }
    },
    {
        "id": "FT--37bfc2c-STPMP00",
        "vduId": "stp",
        "changeType": "ADDED",
        "computeResource": {
            "vimConnectionId": "vimid",
            "resourceId": "db0fa23f-d4d1-435e-ac6f-9884a3520b3b"
        }
    },
    {
        "id": "FT--37bfc2c-STPMP01",
        "vduId": "stp",
        "changeType": "ADDED",
        "computeResource": {
            "vimConnectionId": "vimid",
            "resourceId": "bdce33fc-882b-4160-b575-efa2b6a6e129"
        }
    }
],
"links": {
    "self": {
        "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/vnf_lcm_op_occs/lcmOp-9cd65a60-6731-4e1e-8ec1-8db670d910ea"
    },
    "vnfInstance": {

```

```

        "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/
vnflcm/v1/vnf_instances/dsrSignaling_DRA-7ecac352-112a-41e1-9d15-1f0c5b789b6f"
    },
    "isAutomaticInvocation": false,
    "isCancelPending": false
},
{
    "id": "lcmOp-fbdb7855-db02-418b-8859-1a0b4ecc5f30",
    "operationState": "PROCESSING",
    "stateEnteredTime": "2023/07/08 10:25:32 GMT",
    "startTime": "2023/07/08 10:22:50 GMT",
    "vnfInstanceId": "dsrSignaling_DRA-8572edff-2822-41e0-b2a7-b82ef18f73d9",
    "operation": "INSTANTIATE",
    "operationParams": {
        "flavourId": "DIAMETER",
        "instantiationLevelId": "small",
        "extVirtualLinks": "extVirtualLinks",
        "extManagedVirtualLinks": [
            {
                "resourceId": "10.75.189.241"
            },
            {
                "resourceId": "10.75.189.196"
            }
        ],
        "vimConnectionInfo": {
            "MvlCloud": {
                "vimId": "vimid",
                "vimType": "OpenStack",
                "interfaceInfo": {
                    "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
                },
                "accessInfo": {
                    "username": "dsrvnfm",
                    "password": "YXV0b21hdGlvbg==",
                    "userDomain": "default",
                    "projectDomain": "default",
                    "project": "VNFM_FT1",
                    "region": "VNFM_FT1"
                },
                "extra": {}
            }
        },
        "localizationLanguage": "localizationLanguage",
        "additionalParams": {
            "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
            "xmiNetwork": {
                "name": "ext-net5-dual",
                "vipSubnetName": "ext-net5-ipv6-subnet",
                "subnet": [
                    {
                        "name": "ext-net5-ipv4",
                        "ipVersion": "IPv4"
                    }
                ]
            }
        }
    }
}

```

```

        "name": "ext-net5-ipv6-subnet",
        "ipVersion": "IPv6"
    }
]
},
"imiNetwork": {
    "name": "xsiNetworkDual2",
    "subnet": [
        {
            "name": "xsiNetworkDual2-IPv4",
            "ipVersion": "IPv4"
        },
        {
            "name": "xsiNetworkDual2-IPv6",
            "ipVersion": "IPv6"
        }
    ]
},
"soamAffinityPolicy": "soft-anti-affinity",
"daAffinityPolicy": "soft-anti-affinity",
"ipfeAffinityPolicy": "soft-anti-affinity",
"stpAffinityPolicy": "soft-anti-affinity",
"sbrAffinityPolicy": "soft-anti-affinity",
"udrAffinityPolicy": "soft-anti-affinity",
"xsiNetwork": [
    {
        "name": "xsiNetworkDual",
        "subnet": [
            {
                "name": "xsiIPv4",
                "ipVersion": "IPv4"
            },
            {
                "name": "xsiIPv6",
                "ipVersion": "IPv6"
            }
        ]
    }
],
{
    "name": "xsiNetworkDual2",
    "subnet": [
        {
            "name": "xsiNetworkDual2-IPv4",
            "ipVersion": "IPv4"
        },
        {
            "name": "xsiNetworkDual2-IPv6",
            "ipVersion": "IPv6"
        }
    ]
}
],
"primaryNoamVmName": "FT--8bedbd47-DN000",
"noamSgName": "FT__8bedbd47_DN0SG"
}
},

```

```

"resourceChanges": {
  "affectedVnfc": [
    {
      "id": "FT--6a21a54c-DS000",
      "vduId": "soam",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "c2150a18-90c1-4c87-848e-9747d79c56f8"
      }
    },
    {
      "id": "FT--6a21a54c-DS001",
      "vduId": "soam",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "926d7fd9-496f-43d9-ae46-259dc25ee074"
      }
    },
    {
      "id": "FT--6a21a54c-DMP00",
      "vduId": "damp",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "748bb2f1-fe85-4736-854f-8a2eca14bd7f"
      }
    },
    {
      "id": "FT--6a21a54c-DMP01",
      "vduId": "damp",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "cb72d77f-df03-4514-88ed-068c28847f48"
      }
    },
    {
      "id": "FT--6a21a54c-DIP00",
      "vduId": "ipfe",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "32257af4-284c-4d44-b953-d561de6ca076"
      }
    },
    {
      "id": "FT--6a21a54c-DIP01",
      "vduId": "ipfe",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "44e4bf34-36a7-4db1-9459-ce8241e45cc6"
      }
    }
  ]
}

```

```

    ]
  },
  "links": {
    "self": {
      "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/
vnflcm/v1/vnf_lcm_op_occs/lcmOp-fbdb7855-db02-418b-8859-1a0b4ecc5f30"
    },
    "vnfInstance": {
      "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/
vnflcm/v1/vnf_instances/dsrSignaling_DRA-8572edff-2822-41e0-b2a7-b82ef18f73d9"
    }
  },
  "isAutomaticInvocation": false,
  "isCancelPending": false
},
{
  "id": "lcmOp-8b44dfa9-839e-439e-baa7-6d6bd8da247c",
  "operationState": "COMPLETED",
  "stateEnteredTime": "2023/07/08 10:16:13 GMT",
  "startTime": "2023/07/08 09:53:14 GMT",
  "vnfInstanceId": "dsrNetworkOam_DRA-404255ac-a565-4549-9665-fde586feac99",
  "operation": "INSTANTIATE",
  "operationParams": {
    "flavourId": "DSR NOAM",
    "instantiationLevelId": "HA",
    "extVirtualLinks": "extVirtualLinks",
    "extManagedVirtualLinks": [],
    "vimConnectionInfo": {
      "MvlCloud": {
        "vimId": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
        }
      },
      "accessInfo": {
        "username": "dsrvnfm",
        "password": "YXV0b21hdGlvbg==",
        "userDomain": "default",
        "projectDomain": "default",
        "project": "VNFM_FT1",
        "region": "VNFM_FT1"
      }
    },
    "extra": {}
  }
},
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
    "xmiNetwork": {
      "name": "ext-net",
      "vipSubnetName": "ext-net-ipv6-subnet",
      "subnet": [
        {
          "name": "ext-net-subnet",
          "ipVersion": "IPv4"
        }
      ]
    }
  }
}

```

```

        {
          "name": "ext-net-ipv6-subnet",
          "ipVersion": "IPv6"
        }
      ]
    },
    "imiNetwork": {
      "name": "imiNetworkDual",
      "subnet": [
        {
          "name": "test11",
          "ipVersion": "IPv4"
        },
        {
          "name": "test6",
          "ipVersion": "IPv6"
        }
      ]
    }
  }
},
"resourceChanges": {
  "affectedVnfc": [
    {
      "id": "FT--8bedbd47-DN000",
      "vduId": "noam",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "4e77e775-2d4d-4246-ac02-e83997217087"
      }
    },
    {
      "id": "FT--8bedbd47-DN001",
      "vduId": "noam",
      "changeType": "ADDED",
      "computeResource": {
        "vimConnectionId": "vimid",
        "resourceId": "5b808545-1bbe-4e10-beff-bce21650301f"
      }
    }
  ]
},
"links": {
  "self": {
    "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/vnf_lcm_op_occs/lcmOp-8b44dfa9-839e-439e-baa7-6d6bd8da247c"
  },
  "vnfInstance": {
    "href": "https://[2606:b400:605:b818:6e41:6aff:fec7:8380]:8443/vnflcm/v1/vnf_instances/dsrNetwork0am_DRA-404255ac-a565-4549-9665-fde586feac99"
  }
},
"isAutomaticInvocation": false,
"isCancelPending": false

```

```
}  
]
```

15

Terminating a VNF

This procedure represents the **Terminate VNF** operation. The client can use this procedure to terminate a VNF instance. The POST method terminates a VNF instance.

Following are the two types of request parameters for the **Terminate VNF** operation:

- **FORCEFUL** : The VNFM deletes the VNF and releases the resources immediately after accepting the request.
- **GRACEFUL** : After accepting the request, the VNFM first validates if the VNF configuration is cleaned up. Once the validation is successful, VNFM deletes the VNF and releases the resources.

Note

VNFM does not support clean-up or reverse cloud-init. The user must manually clean the configuration before Graceful Termination.

Below table describes the parameters used for sending request to VNFM.

Table 15-1 Parameters and Definitions for Terminating VNF

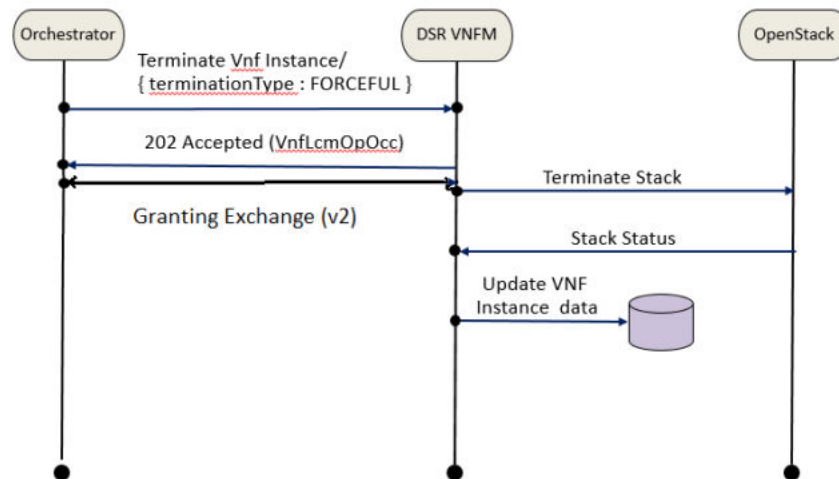
Parameters	Definitions
terminationType	Indicates whether forceful or graceful termination is requested.

15.1 Forceful Termination

The VNFM will delete the VNF immediately after accepting the request. The instance file is updated with VNF Operational State set to **STOPPED**.

Note

If the VNF is still in service, requesting forceful termination can adversely impact the network service.

Figure 15-1 Forceful termination

Terminating DSR and SDS VNF Instance Forcefully

Sample Request:

Request URL: POST: `https://<VNFM HOST IP>:8443/vnflcm/<vnfm_version>/vnf_instances/< VNF ID received from create request>/terminate`

Accept: `application/json`

Content-Type: `application/json`

Authorization: Token generated after login

```
{
  "terminationType": "FORCEFUL"
}
```

Sample Response

Response Code: 202

```
{
  location: https://localhost:8443/vnflcm/<vnfm_version>/vnf_lcm_op_occs/
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6
  date: Tue, 29 Jan 2019 10:39:24 GMT
  content-length: 0  content-type:
  application/xml
}
```

15.2 Graceful Termination

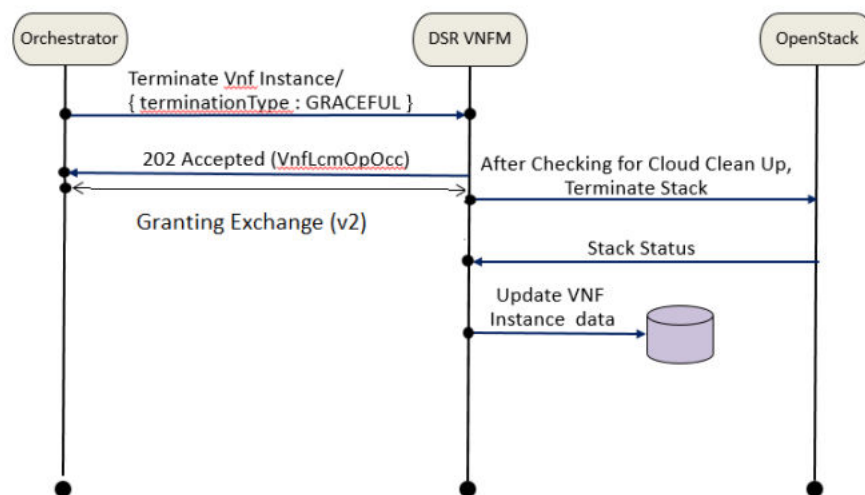
The VNFM first validates that the VNF configuration is cleaned up after accepting the request. Once the configuration is cleaned, the VNFM deletes the VNF. The instance file is updated, setting the VNF Operational State to **STOPPED**.

This cleanup process is automated by the VNFM for DSR, SDS Signaling, and DR NOAM from NOAM. If Appworks configuration are not cleaned up automatically by VNFM, the orchestrator's graceful termination fails.

Note

- VNFM must automatically clean up AppWorks configurations before initiating graceful termination.
- `gracefulTerminationTimeout` specifies the sleep time before graceful stack termination.

Figure 15-2 Graceful termination



Sample Request for graceful termination::

Terminating VNF Instance gracefully

Request URL: POST: `https://<<VNFM HOST IP>>:8443/vnfm/<<vnfm_version>>/vnf_instances/< VNF ID received from create request>/terminate`

Accept: `application/json`

Content-Type: `application/json`

20.3.1.2.1. Authorization : `<Token generated after login>`

```
{
  "terminationType": "GRACEFUL",
  "gracefulTerminationTimeout": 0
}
```

20.3.1.2.2.

Sample Response for graceful termination:

Response Body for Terminate VNF Instance

Response Code : 202

```
{  
    location: https://localhost:8443/vnflcm/(<vnfm_version>)/vnf_lcm_op_occs/  
lcmOp-fb21f9d3-43ad-46cd-a03f-7220bb36a5c6  
  
    date: Tue, 29 Jan 2019 10:39:24 GMT  
  
    content-length: 0    content-type:  
    application/xml  
}
```

Segregation of OAM and Replication Traffic Support (VNFM - V1)

This feature supports the following:

- Separate network for OAM and Replication traffic
- Supports DSR NOAM and Signaling VNFs.

Replication traffic is optional for a user where it can be used as either Inter-NE network (xmiRepl) or Intra-NE network (imiRepl) or Both or None. xmiRepl/imiRepl and xmi/imi supports Single, Dual, and Mix mode subnet, where both OAM and Replication service network can be on IPV4 or IPV6.

For Signaling VNF, it supports for all the following flavours:

- DIAMETER
- SBR
- UDR
- STP

In the scenario where both VNFM and VIM networks are in IPv6, user should ensure that both NOAM and Signaling have same replication network.

The user can configure the new xmiRepl and imiRepl in the same or different interface based on NOAM and Signaling VNFs.

While instantiating with xmiRepl/imiRepl, inside these the network address of the subnets are to be provided for the networks being used.

The following details must be provided only if the replication networks are in different cloud and not overlapping subnets or it is applicable only for remote network addresses and not for local network addresses:

- Noam contains - soamReplNetworkAddress/soamImiReplNetworkAddress ,
drNoamReplNetworkAddress/drNoamImiReplNetworkAddress
- Signaling contains - noamReplNetworkAddress/noamImiReplNetworkAddress ,
drNoamReplNetworkAddress/drNoamImiReplNetworkAddress
- DR-Noam contains - noamReplNetworkAddress/noamImiReplNetworkAddress

VNFM configures **Services** under **Networking**, based on the following user inputs:

- If user input contains both xmiRepl and imiRepl.

Main Menu: Configuration -> Networking -> Services

Fri Jul

Name	Intra-NE Network	Inter-NE Network
OAM	imi	xmi
Replication	imiRepl	xmiRepl
Signaling	Unspecified	Unspecified
HA_Secondary	Unspecified	Unspecified
HA_MP_Secondary	Unspecified	Unspecified
Replication_MP	imi	Unspecified
ComAgent	imi	Unspecified

- If user input contains only xmiRepl.

Main Menu: Configuration -> Networking -> Services

Fri Jul

Name	Intra-NE Network	Inter-NE Network
OAM	imi	xmi
Replication	xmiRepl	xmiRepl
Signaling	Unspecified	Unspecified
HA_Secondary	Unspecified	Unspecified
HA_MP_Secondary	Unspecified	Unspecified
Replication_MP	imi	Unspecified
ComAgent	imi	Unspecified

- If user input contains only imiRepl.

Main Menu: Configuration -> Networking -> Services

Fr

Name	Intra-NE Network	Inter-NE Network
OAM	imi	xmi
Replication	imiRepl	imiRepl
Signaling	Unspecified	Unspecified
HA_Secondary	Unspecified	Unspecified
HA_MP_Secondary	Unspecified	Unspecified
Replication_MP	imi	Unspecified
ComAgent	imi	Unspecified

Note

- The networks used for all the interfaces (xmi, imi, imiRepl and xmiRepl) should be different.
- Use only Public Networks for replication interfaces (xmiRepl and imiRepl).
- The replication network address must be passed in the request body only if the replication networks are on different cloud and not overlapping subnets or for remote network addresses.

16.1 Sample request for DYNAMIC IP model of NOAM VNF in Dual Subnet

Following is the sample request for Segregation of OAM and Replication Traffic for DYNAMIC IP model of NOAM VNF (Dual Subnet):

Instantiating NOAM Request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<VNF ID received from create request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```
{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
      "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
          "ipVersion": "IPv6",
          "name": "ext-net5-ipv6-subnet"
        },
        {
          "ipVersion": "IPv4",
          "name": "ext-net5-ipv4"
        }
      ]
    },
    "imiNetwork": {
      "name": "imiNetworkDual",
      "subnet": [{
```

```

        "ipVersion": "IPv6",
        "name": "test6"
    },
    {
        "ipVersion": "IPv4",
        "name": "test11"
    }
]
},
"xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
    },
    {
        "ipVersion": "IPv4",
        "name": "ext-net-subnet"
    }
]
},
"imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6"
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual3-IPv4"
    }
]
},
"ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
"flavor": "dsr.noam",
"image": "DSR-9.0.2.0.0.vmdk",
"availabilityZone": "nova",
"noamAffinityPolicy": "anti-affinity",
"soamReplNetworkAddress": ["2606:b400:605:b818::/64",
"10.75.189.128/25"],
"soamImiReplNetworkAddress": ["2001:db8:1225::/64",
"192.168.18.0/24"],
"drNoamReplNetworkAddress": ["2606:b400:605:b849::/64",
"10.75.217.0/24"],
"drNoamImiReplNetworkAddress": ["2001:db8:1227::/64",
"192.168.20.0/24"]
}
}

```

Note

vipSubnetName field is to be used only in case of Dual Subnet.

16.2 Sample request for DYNAMIC IP model of NOAM VNF in Single Subnet

Following is the sample request for request for Segregation of OAM and Replication Traffic for DYNAMIC IP model of NOAM VNF (Single Subnet)

Instantiating NOAM Request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<VNF ID received from create request>/instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```
{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
      "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
          "ipVersion": "IPv6",
          "name": "ext-net5-ipv6-subnet"
        }]
      },
      "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [{
          "ipVersion": "IPv6",
```

```

        "name": "test6"
      }]
    },
    "xmiReplNetwork": {
      "name": "ext-net",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
      }]
    },
    "imiReplNetwork": {
      "name": "xsiNetworkDual3",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6"
      }]
    },
    "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
    "flavor": "dsr.noam",
    "image": "DSR-9.0.2.0.0.vmdk",
    "availabilityZone": "nova",
    "noamAffinityPolicy": "anti-affinity",
    "soamReplNetworkAddress": ["2606:b400:605:b818::/64"],
    "soamImiReplNetworkAddress": ["2001:db8:1225::/64"],
    "drNoamReplNetworkAddress": ["2606:b400:605:b849::/64"],
    "drNoamImiReplNetworkAddress": ["2001:db8:1227::/64"]
  }
}

```

16.3 Sample request for FIXED IP model of NOAM VNF in Dual Subnet

Following is a sample for Segregation of OAM and Replication Traffic for FIXED IP model of NOAM VNF (Dual Subnet).

Instantiating NOAM Request generated

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<VNF ID received from create request > request>/instantiate`

Accept: `application/json`

Content-Type: `application/json`

Authorization : `<Token generated after login>`

```

{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {

```

```

        "endpoint": "https://mv1-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
    }
}
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "ext-net5-ipv6-subnet",
            "fixedIps": {
                "primaryNoamIp": "2001:db8:1221::341",
                "secondaryNoamIp": "2001:db8:1221::193",
                "noamVip": "2001:db8:1221::194"
            }
        }
    ],
    {
        "ipVersion": "IPv4",
        "name": "ext-net5-ipv4",
        "fixedIps": {
            "primaryNoamIp": "10.75.189.224",
            "secondaryNoamIp": "10.75.189.225"
        }
    }
]
},
"imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6",
        "fixedIps": {
            "primaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80e0",
            "secondaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80f9"
        }
    }
],
    {
        "ipVersion": "IPv4",
        "name": "test11",
        "fixedIps": {
            "primaryNoamImiIp": "10.75.189.224",

```

```

        "secondaryNoamImiIp": "10.75.189.236"
    }
}
],
"xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet",
        "fixedIps": {
            "primaryNoamReplXmiIp": "2001:db8:1221::341",
            "secondaryNoamReplXmiIp": "2001:db8:1221::193"
        }
    }
},
{
    "ipVersion": "IPv4",
    "name": "ext-net-subnet",
    "fixedIps": {
        "primaryNoamReplXmiIp": "192.168.2.1",
        "secondaryNoamReplXmiIp": "192.168.2.2"
    }
}
]
},
"imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6",
        "fixedIps": {
            "primaryNoamReplImiIp": "2001:db8:1244::1ab",
            "secondaryNoamReplImiIp": "2001:db8:1244::1cf"
        }
    }
},
{
    "ipVersion": "IPv4",
    "name": "xsiNetworkDual3-IPv4",
    "fixedIps": {
        "primaryNoamReplImiIp": "10.75.23.1",
        "secondaryNoamReplImiIp": "10.75.23.2"
    }
}
]
},
"ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
"flavor": "dsr.noam",
"image": "DSR-9.0.2.0.0.vmdk",
"availabilityZone": "nova",
"noamAffinityPolicy": "anti-affinity",

```

```

        "soamReplNetworkAddress": ["2606:b400:605:b818::/64",
        "10.75.189.128/25"],
        "soamImiReplNetworkAddress": ["2001:db8:1225::/64",
        "192.168.18.0/24"],
        "drNoamReplNetworkAddress": ["2606:b400:605:b849::/64",
        "10.75.217.0/24"],
        "drNoamImiReplNetworkAddress": ["2001:db8:1227::/64",
        "192.168.20.0/24"]
    }
}

```

① Note

"vipSubnetName" field has to be used only in case of Dual Subnet.

16.4 Sample request for FIXED IP model of NOAM VNF in Single Subnet

Following is a sample request for Segregation of OAM and Replication Traffic for FIXED IP model of DSR NOAM VNF (Single Subnet).

Instantiating the NOAM Request generated

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/<VNF ID received from create request>/instantiate`

Accept: `application/json`

Content-Type: `application/json`

Authorization : `<Token generated after login>`

```

{
  "flavourId": "DSR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    }
  },
  "accessInfo": {
    "project": "*****",
    "region": "RegionOne",
    "username": "*****",
    "password": "*****",
    "userDomain": "Default",
    "projectDomain": "default"
  }
}

```

```

"localizationLanguage": "localizationLanguage",
"additionalParams": {
  "xmiNetwork": {
    "name": "ext-net5-dual",
    "vipSubnetName": "ext-net5-ipv6-subnet",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext-net5-ipv6-subnet",
      "fixedIps": {
        "primaryNoamIp": "2606:b400:605:b818:6e41:6aff:fec7:80e0",
        "secondaryNoamIp":
"2606:b400:605:b818:6e41:6aff:fec7:80f9",
        "noamVip": "2606:b400:605:b818:6e41:6aff:fec7:80fc"
      }
    }]
  },
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "test6",
      "fixedIps": {
        "primaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:8010",
        "secondaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:8019"
      }
    }]
  },
  "xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext-net-ipv6-subnet",
      "fixedIps": {
        "primaryNoamReplXmiIp": "2001:db8:1221::341",
        "secondaryNoamReplXmiIp": "2001:db8:1221::193"
      }
    }]
  },
  "imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "xsiNetworkDual3-IPv6",
      "fixedIps": {
        "primaryNoamReplImiIp": "2001:db8:1244::1ab",
        "secondaryNoamReplImiIp": "2001:db8:1244::1cf"
      }
    }]
  },
  "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",

```

```

        "flavor": "dsr.noam",
        "image": "DSR-9.0.2.0.0.vmdk",
        "availabilityZone": "nova",
        "noamAffinityPolicy": "anti-affinity",
        "soamReplNetworkAddress": ["2606:b400:605:b818::/64"],
        "soamImiReplNetworkAddress": ["2001:db8:1225::/64"],
        "drNoamReplNetworkAddress": ["2606:b400:605:b849::/64"],
        "drNoamImiReplNetworkAddress": ["2001:db8:1227::/64"]
    }
}

```

16.5 Sample request for Signaling flavor DIAMETER with multiple xsi for dynamic IP in Dual Subnet deployment model

Following is a sample request for Segregation of OAM and Replication Traffic for Signaling flavor DIAMETER with multiple xsi (1,2,4 xsi interface) for dynamic IP (Dual Subnet) deployment model.

Instantiating the first signaling VNF request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****"
      }
    }
  }
}

```

```

        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
    }
}
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "ext-net5-ipv6-subnet"
        },
        {
            "ipVersion": "IPv4",
            "name": "ext-net5-ipv4"
        }
    ]
},
"imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6"
    },
    {
        "ipVersion": "IPv4",
        "name": "test11"
    }
]
},
"xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet"
    },
    {
        "ipVersion": "IPv4",
        "name": "ext-net-subnet"
    }
]
},
"imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6"
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual3-IPv4"
    }
]
}
]

```

```

    },
    "xsiNetwork": [{
        "name": "ext-net2",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "xsiIPv6"
        },
        {
            "ipVersion": "IPv4",
            "name": "xsiIPv4"
        }
    ]
},
{
    "name": "xsiNetworkDual2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual2-IPv6"
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual2-IPv4"
    }
    ]
}
],
"ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
"primaryNoamVmName": "NOAM00-32cd6138",
"noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0.vmdk",
"soamAvailabilityZone": "nova",
"ipfeFlavor": "dsr.ipfe",
"ipfeImage": "DSR-9.0.2.0.0.vmdk",
"ipfeAvailabilityZone": "nova",
"daFlavor": "dsr.da",
"daImage": "DSR-9.0.2.0.0.vmdk",
"daAvailabilityZone": "nova",
"stpFlavor": "dsr.stp",
"stpImage": "DSR-9.0.2.0.0.vmdk",
"stpAvailabilityZone": "nova",
"soamAffinityPolicy": "anti-affinity",
"ipfeAffinityPolicy": "anti-affinity",
"daAffinityPolicy": "soft-anti-affinity",
"noamReplNetworkAddress": ["2606:b400:605:b84a::/64",
"10.75.234.0/25"],
"noamImiReplNetworkAddress": ["2001:db8:1224::/64",
"192.168.25.0/24"],
"drNoamReplNetworkAddress": ["2606:b400:605:b849::/64",
"10.75.217.0/24"],
"drNoamImiReplNetworkAddress": ["2001:db8:1227::/64",
"192.168.20.0/24"]
}
}

```

Note

"vipSubnetName" field is to be used only in case of Dual Subnet.

vnfConfigurableProperties:

```
{
  "ntpServerIp": "10.250.32.10",
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "name": "test11",
      "ipVersion": "IPv4"
    },
    {
      "name": "test6",
      "ipVersion": "IPv6"
    }
  ]
},
  "xsiNetworkCount": 4,
  "primaryNoamVmName": "csarNo-DN000",
  "noamSgName": "csarNo_DNOSG",
  "noamXmiIps": [
    "10.75.189.150",
    "10.75.189.212"
  ]
}
```

16.6 Sample request for Signaling flavor DIAMETER with multiple xsi for dynamic IP in Single Subnet deployment model

Following is a sample request for Segregation of OAM and Replication Traffic for DSR Signaling flavor DIAMETER with multiple xsi (1,2,4 xsi interface) for dynamic IP (Single Subnet) deployment model.

Instantiating the first signaling VNF request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>
/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```
{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
}
```

```

"extManagedVirtualLinks": [{
  "id": "",
  "virtualLinkDescId": "active NOAM",
  "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
},
{
  "id": "",
  "virtualLinkDescId": "standby NOAM",
  "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
}
],
"vimConnectionInfo": {
  "MvlCloud": {
    "vimId": "vimId",
    "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
    "interfaceInfo": {
      "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default"
    }
  }
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
  "xmiNetwork": {
    "name": "ext-net5-dual",
    "vipSubnetName": "ext-net5-ipv6-subnet",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext-net5-ipv6-subnet"
    }]
  },
  "imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "test6"
    }]
  },
  "xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext-net-ipv6-subnet"
    }]
  },
  "imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
      "ipVersion": "IPv6",

```

```

        "name": "xsiNetworkDual3-IPv6"
    }
  ],
  "xsiNetwork": [{
    "name": "ext-net2",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "xsiIPv6"
    }]
  }],
  {
    "name": "xsiNetworkDual2",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "xsiNetworkDual2-IPv6"
    }]
  }
],
"ntpServerIp": "10.250.32.10",
"primaryNoamVmName": "NOAM00-32cd6138",
"noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0.vmdk",
"soamAvailabilityZone": "nova",
"ipfeFlavor": "dsr.ipfe",
"ipfeImage": "DSR-9.0.2.0.0.vmdk",
"ipfeAvailabilityZone": "nova",
"daFlavor": "dsr.da",
"daImage": "DSR-9.0.2.0.0.vmdk",
"daAvailabilityZone": "nova",
"stpFlavor": "dsr.stp",
"stpImage": "DSR-9.0.2.0.0.vmdk",
"stpAvailabilityZone": "nova",
"soamAffinityPolicy": "anti-affinity",
"ipfeAffinityPolicy": "anti-affinity",
"daAffinityPolicy": "soft-anti-affinity",
"stpAffinityPolicy": "soft-anti-affinity",
"noamReplNetworkAddress": ["2606:b400:605:b84a::/64"],
"noamImiReplNetworkAddress": ["2001:db8:1224::/64"],
"drNoamReplNetworkAddress": ["2606:b400:605:b849::/64"],
"drNoamImiReplNetworkAddress": ["2001:db8:1227::/64"]
}
}

```

16.7 Sample request for Signaling flavor DIAMETER with multiple xsi for fixed IP in Dual Subnet deployment model

Following is a sample request for Segregation of OAM and Replication Traffic for DSR Signaling flavor DIAMETER with multiple xsi (1,2,4 xsi interface) for Fixed IP (Dual Subnet) deployment model.

Instantiating the first signaling VNF request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```
{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
      }
    },
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
      "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
          "ipVersion": "IPv6",
          "name": "ext-net5-ipv6-subnet",
          "fixedIps": {
            "primarySoamXmiIp": "2001:db8:1221::254",
            "secondarySoamXmiIp": "2001:db8:1221::2fe",
            "soamVip": "2606:b400:605:b813::3c",
            "dampXmiIps": ["2001:db8:1221::255",
              "2001:db8:1221::100"],
          }
        }
      }
    }
  }
}
```

```

        "ipfeXmiIps": ["2001:db8:1221::22a",
"2001:db8:1221::2ca"]
    },
    {
        "ipVersion": "IPv4",
        "name": "ext-net5-ipv4",
        "fixedIps": {
            "primarySoamXmiIp": "10.75.189.54",
            "secondarySoamXmiIp": "10.75.189.55",
            "dampXmiIps": ["10.75.189.56", "10.75.189.57"],
            "ipfeXmiIps": ["10.75.189.58", "10.75.189.59"]
        }
    }
]
},
"imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6",
        "fixedIps": {
            "primarySoamImiIp": "2001:db8:1234::2d9",
            "secondarySoamImiIp": "2001:db8:1234::3c5",
            "dampImiIps": ["2001:db8:1234::162",
"2001:db8:1234::27"],
            "ipfeImiIps": ["2001:db8:1234::109",
"2001:db8:1234::161"]
        }
    },
    {
        "ipVersion": "IPv4",
        "name": "test11",
        "fixedIps": {
            "primarySoamImiIp": "10.75.189.191",
            "secondarySoamImiIp": "10.75.189.192",
            "dampImiIps": ["10.75.189.193", "10.75.189.194"],
            "ipfeImiIps": ["10.75.189.195", "10.75.189.196"]
        }
    }
]
},
"xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet",
        "fixedIps": {
            "primarySoamReplXmiIp": "2001:db8:1221::254",
            "secondarySoamReplXmiIp": "2001:db8:1221::2fe",
            "dampReplXmiIps": ["2001:db8:1221::255",
"2001:db8:1221::100"],

```

```

        "ipfeReplXmiIps": ["2001:db8:1221::22a",
"2001:db8:1221::2ca"]
    },
    {
        "ipVersion": "IPv4",
        "name": "ext-net-subnet",
        "fixedIps": {
            "primarySoamReplXmiIp": "10.75.291.1",
            "secondarySoamReplXmiIp": "10.75.291.2",
            "dampReplXmiIps": ["10.75.291.3", "10.75.291.4"],
            "ipfeReplXmiIps": ["10.75.291.5", "10.75.291.6"]
        }
    }
]
},
"imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6",
        "fixedIps": {
            "primarySoamReplImiIp": "2001:db8:1244::26f",
            "secondarySoamReplImiIp": "2001:db8:1244::d9",
            "dampReplImiIps": ["2001:db8:1244::2d8",
"2001:db8:1244::7c"],
            "ipfeReplImiIps": ["2001:db8:1244::37a",
"2001:db8:1244::a9"]
        }
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiNetworkDual3-IPv4",
        "fixedIps": {
            "primarySoamReplImiIp": "192.168.75.1",
            "secondarySoamReplImiIp": "192.168.75.2",
            "dampReplImiIps": ["192.168.75.3", "192.168.75.4"],
            "ipfeReplImiIps": ["192.168.75.5", "192.168.75.6"]
        }
    }
]
},
"xsiNetwork": [{
    "name": "ext-net2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiIPv6",
        "fixedIps": {
            "dampXsiIps": ["fd0d:deba:d97c:2d::33c",
"fd0d:deba:d97c:2d::411"],
            "ipfeXsiIps": ["fd0d:deba:d97c:2d::253",
"fd0d:deba:d97c:2d::27b"]
        }
    }
]
}

```

```

        }
    },
    {
        "ipVersion": "IPv4",
        "name": "xsiIPv4",
        "fixedIps": {
            "dampXsiIps": ["192.163.41.1", "192.163.41.2"],
            "ipfeXsiIps": ["192.163.41.3", "192.163.41.4"]
        }
    }
],
},
{
    "name": "xsiNetworkDual2",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual2-IPv6",
        "fixedIps": {
            "dampXsiIps": ["fd0d:deba:d97c:2d::33c",
"fd0d:deba:d97c:3d::412"],
            "ipfeXsiIps": ["fd0d:deba:d97c:2d::253",
"fd0d:deba:d97c:3d::27c"]
        }
    }
},
{
    "ipVersion": "IPv4",
    "name": "xsiNetworkDual2-IPv4",
    "fixedIps": {
        "dampXsiIps": ["192.163.42.1", "192.163.42.2"],
        "ipfeXsiIps": ["192.163.42.3", "192.163.42.4"]
    }
}
]
},
    "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
    "primaryNoamVmName": "NOAM00-32cd6138",
    "noamSgName": "dsrNetworkOam_NOAM_32cd6138_SG",
    "soamFlavor": "dsr.soam",
    "soamImage": "DSR-9.0.2.0.0.vmdk",
    "soamAvailabilityZone": "nova",
    "ipfeFlavor": "dsr.ipfe",
    "ipfeImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeAvailabilityZone": "nova",
    "daFlavor": "dsr.da",
    "daImage": "DSR-9.0.2.0.0.vmdk",
    "daAvailabilityZone": "nova",
    "stpFlavor": "dsr.stp",
    "stpImage": "DSR-9.0.2.0.0.vmdk",
    "stpAvailabilityZone": "nova",
    "soamAffinityPolicy": "anti-affinity",
    "ipfeAffinityPolicy": "anti-affinity",

```

```

        "daAffinityPolicy": "soft-anti-affinity",
        "noamReplNetworkAddress": ["2606:b400:605:b84a::/64",
"10.75.234.0/25"],
        "noamImiReplNetworkAddress": ["2001:db8:1224::/64",
"192.168.25.0/24"],
        "drNoamReplNetworkAddress": ["2606:b400:605:b849::/64",
"10.75.217.0/24"],
        "drNoamImiReplNetworkAddress": ["2001:db8:1227::/64",
"192.168.20.0/24"]
    }
}

```

① Note

"vipSubnetName" field is to be used only in case of Dual Subnet.

16.8 Sample request for Signaling flavor DIAMETER with multiple xsi for fixed IP in Single Subnet deployment model

Following is a sample request for Segregation of OAM and Replication Traffic for DSR Signaling flavor DIAMETER with multiple xsi (1,2,4 xsi interface) for Fixed IP (Single Subnet) deployment model.

Instantiating the first signaling VNF request generated

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>
/vnf_instances/ < VNF ID received from create request > /instantiate

Accept: application/json

Content-Type: application/json

Authorization : <Token generated after login>

```

{
  "flavourId": "DIAMETER",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "8a4d1ec6-367a-4b1a-978d-2c4eae3daec3"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "2bed5886-8c97-4623-8da3-9c500cce71e3"
  }
  ],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {

```

```

        "endpoint": "https://mv1-dev1.us.oracle.com:5000/v3"
    },
    "accessInfo": {
        "project": "*****",
        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
    }
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "ext-net5-ipv6-subnet",
            "fixedIps": {
                "primarySoamXmiIp": "2001:db8:1221::254",
                "secondarySoamXmiIp": "2001:db8:1221::2fe",
                "soamVip": "2606:b400:605:b813::3c",
                "dampXmiIps": ["2001:db8:1221::255",
"2001:db8:1221::100"],
                "ipfeXmiIps": ["2001:db8:1221::22a", "2001:db8:1221::2ca"]
            }
        }
    ]
},
    "imiNetwork": {
        "name": "imiNetworkDual",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "test6",
            "fixedIps": {
                "primarySoamImiIp": "2001:db8:1234::2d9",
                "secondarySoamImiIp": "2001:db8:1234::3c5",
                "dampImiIps": ["2001:db8:1234::162", "2001:db8:1234::27"],
                "ipfeImiIps": ["2001:db8:1234::109", "2001:db8:1234::161"]
            }
        }
    ]
},
    "xmiReplNetwork": {
        "name": "ext-net",
        "subnet": [{
            "ipVersion": "IPv6",
            "name": "ext-net-ipv6-subnet",
            "fixedIps": {
                "primarySoamReplXmiIp": "2001:db8:1221::254",
                "secondarySoamReplXmiIp": "2001:db8:1221::2fe",
                "dampReplXmiIps": ["2001:db8:1221::255",
"2001:db8:1221::100"],
                "ipfeReplXmiIps": ["2001:db8:1221::22a",

```

```

"2001:db8:1221::2ca"]
    }

    ]]

  },
  "imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "xsiNetworkDual3-IPv6",
      "fixedIps": {
        "primarySoamReplImiIp": "2001:db8:1244::26f",
        "secondarySoamReplImiIp": "2001:db8:1244::d9",
        "dampReplImiIps": ["2001:db8:1244::2d8",
"2001:db8:1244::7c"],
        "ipfeReplImiIps": ["2001:db8:1244::37a",
"2001:db8:1244::a9"]
      }
    }
  ]
},
"xsiNetwork": [{
  "name": "ext-net2",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "xsiIPv6",
    "fixedIps": {
      "dapXsiIps": ["fd0d:deba:d97c:2d::33c",
"fd0d:deba:d97c:2d::411"],
      "ipfeXsiIps": ["fd0d:deba:d97c:2d::253",
"fd0d:deba:d97c:2d::27b"]
    }
  }
}]
},
{
  "name": "xsiNetworkDual2",
  "subnet": [{
    "ipVersion": "IPv6",
    "name": "xsiNetworkDual2-IPv6",

    "fixedIps": {
      "dampXsiIps": ["fd0d:deba:d97c:2d::33c",
"fd0d:deba:d97c:3d::412"],
      "ipfeXsiIps": ["fd0d:deba:d97c:2d::253",
"fd0d:deba:d97c:3d::27c"]
    }
  }
}]
},
"ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
"primaryNoamVmName": "NOAM00-32cd6138",
"noamSgName": "dsrNetwork0am_NOAM_32cd6138_SG",
"soamFlavor": "dsr.soam",
"soamImage": "DSR-9.0.2.0.0.vmdk",

```

```

    "soamAvailabilityZone": "nova",
    "ipfeFlavor": "dsr.ipfe",
    "ipfeImage": "DSR-9.0.2.0.0.vmdk",
    "ipfeAvailabilityZone": "nova",
    "daFlavor": "dsr.da",
    "daImage": "DSR-9.0.2.0.0.vmdk",
    "daAvailabilityZone": "nova",
    "stpFlavor": "dsr.stp",
    "stpImage": "DSR-9.0.2.0.0.vmdk",
    "stpAvailabilityZone": "nova",
    "soamAffinityPolicy": "anti-affinity",
    "ipfeAffinityPolicy": "anti-affinity",
    "daAffinityPolicy": "soft-anti-affinity",
    "stpAffinityPolicy": "soft-anti-affinity",
    "noamReplNetworkAddress": ["2606:b400:605:b84a::/64"],
    "noamImiReplNetworkAddress": ["2001:db8:1224::/64"],
    "drNoamReplNetworkAddress": ["2606:b400:605:b849::/64"],
    "drNoamImiReplNetworkAddress": ["2001:db8:1227::/64"]
  }
}

```

16.9 Sample request for DSR DR NOAM for Dynamic IP in Mix Mode Subnet deployment model

Following is a sample request for Segregation of OAM and Replication Traffic for DSR DR NOAM for Dynamic IP (Mix Mode Subnet) deployment model.

Instantiating the DSR DR NOAM VNF request generated

```

URL: https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>
/vnf_instances/ < VNF ID received from create request > /instantiate
Accept: application/json
Content-Type: application/json
Authorization : <Token generated after login>{
  "flavourId": "DSR DR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "",
    "virtualLinkDescId": "active NOAM",
    "resourceId": "10.75.234.33"
  },
  {
    "id": "",
    "virtualLinkDescId": "standby NOAM",
    "resourceId": "10.75.234.63"
  }
],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      }
    }
  }
}

```

```

    },
    "accessInfo": {
      "project": "*****",
      "region": "RegionOne",
      "username": "*****",
      "password": "*****",
      "userDomain": "Default",
      "projectDomain": "default"
    }
  },
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "xmiNetwork": {
      "name": "ext-net5-dual",
      "vipSubnetName": "ext-net5-ipv6-subnet",
      "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net5-ipv6-subnet"
      },
      {
        "ipVersion": "IPv4",
        "name": "ext-net5-ipv4"
      }
    ]
  },
  "imiNetwork": {
    "name": "imi-private",
    "subnet": [
      {
        "ipVersion": "IPv4",
        "name": "imi-private-sub"
      }
    ]
  },
  "xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "ext-net-ipv6-subnet"
    }]
  },
  "imiReplNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
      "ipVersion": "IPv6",
      "name": "test6"
    },
    {
      "ipVersion": "IPv4",
      "name": "test11"
    }
  ]
}

```

```

    }
  },
  "ntpServerIp": "2606:b400:605:b912:200:5eff:fe00:1f7",
  "primaryNoamVmName": "No852-DN000",
  "noamReplNetworkAddress": ["2606:b400:605:b84a::/64"],
  "noamImiReplNetworkAddress": ["2001:db8:1224::/64", "192.168.25.0/24"]
}

```

① Note

"vipSubnetName" field is to be used only in case of Dual Subnet.

16.10 Sample request for DSR DR NOAM for Fixed IP in Dual Subnet deployment model

Following is a sample request for Segregation of OAM and Replication Traffic for DSR DR NOAM for Fixed IP (Dual Subnet) deployment model.

Instantiating the DSR DR NOAM VNF request generated

URL: `https://<<VNFM HOST IP>>:8443/vnflcm/<<vnfm_version>>/vnf_instances/ < VNF ID received from create request > /instantiate`

Accept: `application/json`

Content-Type: `application/json`

Authorization : `<Token generated after login>`

```

{
  "flavourId": "DSR DR NOAM",
  "instantiationLevelId": "HA",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [{
    "id": "id1",
    "virtualLinkDescId": "Active NOAM IP's",
    "resourceId": "10.75.234.33"
  },
  {
    "id": "id2",
    "virtualLinkDescId": "Secondary NOAM IP's",
    "resourceId": "10.75.234.63"
  }
],
  "vimConnectionInfo": {
    "MvlCloud": {
      "vimId": "vimId",
      "vimType": "ETSINFV.OPENSTACK_KEYSTONE.V_3",
      "interfaceInfo": {
        "endpoint": "https://mvl-dev1.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "project": "*****",

```

```

        "region": "RegionOne",
        "username": "*****",
        "password": "*****",
        "userDomain": "Default",
        "projectDomain": "default"
    }
},
"localizationLanguage": "localizationLanguage",
"additionalParams": {
    "xmiNetwork": {
        "name": "ext-net5-dual",
        "vipSubnetName": "ext-net5-ipv6-subnet",
        "subnet": [{
            "ipVersion": "IPv4",
            "name": "ext-net5-ipv4",
            "fixedIps": {
                "drPrimaryNoamIp": "10.75.234.107",
                "drSecondaryNoamIp": "10.75.234.105"
            }
        },
        {
            "ipVersion": "IPv6",
            "name": "ext-net5-ipv6-subnet",
            "fixedIps": {
                "drPrimaryNoamIp":
"2606:b400:605:b84a:6e41:6aff:fec7:8101",
                "drSecondaryNoamIp":
"2606:b400:605:b84a:6e41:6aff:fec7:83aa",
                "drNoamVip": "2606:b400:605:b84a:6e41:6aff:fec7:818b"
            }
        }
    ]
},
"imiNetwork": {
    "name": "imiNetworkDual",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "test6",
        "fixedIps": {
            "drPrimaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80e0",
            "drSecondaryNoamImiIp":
"2606:b400:605:b818:6e41:6aff:fec7:80f9"
        }
    },
    {
        "ipVersion": "IPv4",
        "name": "test11",
        "fixedIps": {
            "drPrimaryNoamImiIp": "10.75.189.224",
            "drSecondaryNoamImiIp": "10.75.189.236"
        }
    }
}

```

```

        }
    }
}
},
"xmiReplNetwork": {
    "name": "ext-net",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "ext-net-ipv6-subnet",
        "fixedIps": {
            "primaryDrNoamReplXmiIp": "2001:db8:1221::341",
            "secondaryDrNoamReplXmiIp": "2001:db8:1221::193"
        }
    }
},
{
    "ipVersion": "IPv4",
    "name": "ext-net-subnet",
    "fixedIps": {
        "primaryDrNoamReplXmiIp": "192.168.2.1",
        "secondaryDrNoamReplXmiIp": "192.168.2.2"
    }
}
}
},
"imiReplNetwork": {
    "name": "xsiNetworkDual3",
    "subnet": [{
        "ipVersion": "IPv6",
        "name": "xsiNetworkDual3-IPv6",
        "fixedIps": {
            "primaryDrNoamReplImiIp": "2001:db8:1244::1ab",
            "secondaryDrNoamReplImiIp": "2001:db8:1244::1cf"
        }
    }
},
{
    "ipVersion": "IPv4",
    "name": "xsiNetworkDual3-IPv4",
    "fixedIps": {
        "primaryDrNoamReplImiIp": "10.75.23.1",
        "secondaryDrNoamReplImiIp": "10.75.23.2"
    }
}
}
},
"primaryNoamVmName": "FixNo852-DN000",
"ntpServerIp": "10.250.32.10",
"noamReplNetworkAddress": ["2606:b400:605:b84a::/64",
"10.75.234.0/25"],
"noamImiReplNetworkAddress": ["2001:db8:1224::/64", "192.168.25.0/24"]

```

```
}  
}
```

Changing the Default Configurations

This section includes information about changing the default configurations through the following files:

- `VmInfo.yaml`
- `VnfmProperties.yaml`

Images, Availability Zone, and Flavors in VNFM can be configured through rest request of each individual VNF. These are the optional parameters of the rest request. If these optional parameters are not provided in the request body, then VNFM considers the default configurations present in the `VmInfo.yaml` file.

To change default configurations of OAM Network Name for Intra-Ne and Inter-Ne (XMI and IMI) in DSR/SDS:

1. Navigate to `/opt/vnfm/config` path.
2. Edit the below network names in `VnfmProperties.yaml` file.
`<oamInter>xmi</oamInter>`
`<oamIntra>imi</oamIntra>`

Note

Network name should be same for NOAM and signaling VNFs.

To change default configurations of **Replication Network Name for Intra-Ne and Inter-Ne** (XMIRepI and IMIRepI) in DSR:

1. Navigate to `/opt/vnfm/config` path.
2. Edit the below network names in `VnfmProperties.yaml` file.
`<replicationIntra>imiRepl</replicationIntra>`
`<replicationInter>xmiRepl</replicationInter>`

Note

Network name should be same for NOAM and signaling VNFs.

Note

To change default Image, Flavor, Availability zone configurations of VNFM:

- Images, Availability Zone and Flavors in VNFM can be configured through rest request of each individual VNF.
- Flavor, Image and Availability zone are the optional parameter of the rest request.
- If these optional parameters are not given in the request body then VNFM takes the default configurations present in `VmInfo.yaml` file.

17.1 Changing Profile Name

1. Log into the VNFM VM.
2. Change to `/opt/vnfm/config/` folder.
3. Edit `VnfmProperties.yaml`.
4. Find the tag `<dsrHardwareProfileName>`.
5. Change the default name to user defined name.

Note

The user defined image name should be a valid profile name.

Profile/Hardware name

`dsrHardwareProfileName`: DSR Guest

`sdsHardwareProfileName`: SDS Cloud Guest

17.2 Enable VNFM Logs with Different Log Levels (DEBUG, TRACE, WARN, ERROR)

- Open the file `log4j2.xml` located in `/opt/vnfm/config/`
- Replace `level="INFO"` with `level="DEBUG"` (or TRACE or WARN or ERROR) in `<Logger>` tag and save

Note

Default value of level is "INFO"

17.3 Changing the Full Semantic Validation Configuration

Full Semantic Validation includes validation of the following entities passed as a part of the request body:

- Image Names

- Flavor Names

These additional validations delay the processing time of a request and returning a response. The default value for this property is "false".

- Change the value of the property `<fullSemanticValidation>` in the `/opt/vnfm/config/VnfmProperties.yaml` file.

Allowed values can be: true or false.

Subscription Operation (VNFM - V2)

Create Subscription

The following is a sample request for a Create New Subscription Resource:

- Subscriptions API consumer can use this API to subscribe to notifications related to VNF lifecycle management. The data structure contains filtering criteria and a callback URI to which the VNFM subsequently sends notifications about events that match the filter.
- To test the notification endpoint that has been registered by the NFVO as part of the subscription, the VNFM sends a GET request to the notification endpoint URI.
- The NFVO returns a "204 No Content" response to indicate success.
- The VNFM creates a new subscription to notifications related to VNF lifecycle changes, and an "Individual subscription" resource that represents this subscription.
- The VNFM returns "201 Created" response to the NFVO when the subscription has been created.
- The Subscription response contains data representing the "Individual subscription" details created by the VNFM and provides the URI of the newly created resource in the "Location" HTTP header.

Note

- When users delete the VNF instance, it is possible that users might create the new instance with the same instance name. The previous subscription becomes active and starts sending the notification for the same instance if all the filters are matching.
- The attributes "vnfdIds" and "vnfProductsFromProviders" are alternatives to reference to VNF instances that are based on certain VNFDs in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.
- The attributes "vnfInstanceId" and "vnfInstanceNames" are alternatives to reference to particular VNF instances in a filter. They should not be used both in the same filter instance, but one alternative should be chosen.

Query Individual Subscription

The following is a sample request for a Query Individual Subscription resource:

- GET operation to obtain the individual Subscription information by passing the subscription ID.
- The VNFM returns "200 OK" response to the NFVO when the individual subscriptions has been queried.
- The response body contains a representation of the "Individual subscription" resource.

Query All Subscription

The following is a sample request for a Query All Subscription resource:

- GET operation to obtain all Subscription information.
- VNFM returns a "200 OK" response to the NFVO when the list of subscriptions has been queried.

Delete Subscription

Delete operation deletes the individual subscription information, by passing the Subscription ID.

- If the NFVO does not need the subscription anymore, it terminates the subscription by sending a DELETE request to the resource that represents the individual subscription to remove.
- The VNFM returns a "204 No Content" response to the NFVO when the "Individual subscription" resource has been deleted and returns empty response body.

Notification

Precondition: The NFVO has subscribed previously to notifications related to VNF lifecycle management.

- VNFM sends notifications related to VNF lifecycle changes to a subscribed API consumer (NFVO), which has provided the URI of this resource during the subscription process.
- If an event occurs that matches the filtering criteria which are part of the subscription, the VNFM generates a notification that includes information about the event and sends it in the body of a POST request to the URI that the NFVO has registered as part of the subscription request.
- The NFVO acknowledges the successful delivery of the notification by returning a "204 No Content" response.
- If the VNFM does not receive the "204 No Content" response from the NFVO, it can retry sending the notification.

VNFM SNMP ALERTS

- VNFM supports both Single and Dual SNMP Manager for SNMP alerts.
- VNFM acts as a SNMP Agent that generates traps such as system health traps, exception traps, and success notifications.
- VNFM MIB (oracleVnfm.mib) and Top level MIB (tekelec-toplevel-reg.mib) are placed in the "/usr/share/snmp/mibs" directory.
- VNFM generates traps in the following SNMP versions:
 - System health traps - SNMP v2c version
 - VNFM exception and success notifications - SNMP v3 version

Table 19-1 VNFM IP

VNFM IP (ens3)	VNFM IP (ens4)	SNMP Manager 1	SNMP Manager 2	Supported
IPv4	IPv4	IPv4	IPv4	Yes
		IPv6	IPv6	NA
		IPv6	IPv4	NA
		IPv4	IPv6	NA
IPv6	IPv4	IPv4	IPv4	Yes
		IPv4	IPv6	Yes
		IPv6	IPv4	Yes
		IPv6	IPv6	Yes

For more information on Alerts and MIB files, see [VNFM Alarms](#) and [VNFM MIB](#) for more information.

Note

The SNMP receiver IP address should always be reachable from the VNFM server.

Steps to change the SNMP Trap Receiver or Manager

To change the SNMP Trap Receiver or Manager:

1. Edit the IP and port of SNMP Trap Receiver or Manager by changing the property `<address>ip/port</address>` inside the `SnmpReceiverIPs` node of the `VnfmProperties.yaml` file located in the `/opt/vnfm/config/VnfmProperties.yaml` folder.
2. For new ports other than 162, add rule for the specific port in Security Group of VNFM stack. See **Steps to change from Single to Dual SNMP Manager**:

EX:
SnmpReceiverIPs:

```
address:
- 2606:b400:605:b813::5/7400
```

3. Run the following script in dsrvnfm user mode:

```
/var/vnfm/prometheus/snmp_notifier/restart_SnmpNotifier.sh <VNFM IP
Address>
```

The output reflects that the SNMP notifier successfully stopped and started again with the given SNMP Trap Receiver or Manager.

Steps to change from Single to Dual SNMP Manager

To convert from Single to Dual SNMP Manager: For second SNMP Manager receiver port other than 162, add rule in Security group in the following format of VNFM stack.

1. Add second address inside SnmpReceiverIPs property as <address>ip/port</address> in /opt/vnfm/config/VnfmProperties.yaml.
For example:

```
EX:
      SnmpReceiverIPs:
        address: [10.75.189.151/8900, 2606:b400:605:b813::5/7400]
```

2. For second SNMP Manager receiver port other than 162, add rule in Security group in the following format of VNFM stack.

Figure 19-1 SNMP Manager

☐	Egress	IPv6	UDP	8900	:::0	-	Delete Rule
☐	Egress	IPv4	UDP	8900	0.0.0.0/0	-	Delete Rule

3. In dsrvnfm user mode, run the following script:

```
/var/vnfm/prometheus/snmp_notifier/restart_SnmpNotifier.sh <VNFM IP
Address>
```

The output reflects that the SNMP notifier successfully stopped and started again with the given SNMP Trap Receiver or Manager.

SNMP System Traps Configurations

SNMP System traps have some default configurations specified in the file:

```
/var/vnfm/prometheus/alertmanager/alertmanager.yaml
```

Default configurations

```
# The labels by which incoming alerts are grouped together.
```

```
route:
```

```
group_by: ['alertname']
group_wait: 10s
group_interval: 5m
repeat_interval: 30m
receiver: 'web.hook'
```

In order to change the wait time or repeat interval, follow these steps:

In dsrvnfm user mode:

- Edit the group_interval or repeat_interval time configuration in this file:

```
/var/vnfm/prometheus/alertmanager/alertmanager.yaml
```

- Run the following script:

```
/var/vnfm/prometheus/snmp_notifier/restart_SnmpNotifier.sh <VNFM IP Address>
```

19.1 VNFM Alarms

This section includes information about VNFM alarms.

Example OID: 1.3.6.1.4.1.323.5.3.33.1.2.1.3001

Table 19-2 General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
3001	GEN_001	vnfmIllegalArgumentGenAlertNotification	Exception for Illegal argument	Minor
3002	GEN_002	vnfmNullPointerAlertNotification	Exception for a Null Pointer	Minor
3003	GEN_003	vnfmWorkingDirectoryErrorAlertNotification	Error while creating the NSA Directory Fails	Minor
3004	GEN_004	vnfmHttpClientHandlingErrorAlertNotification	Error when there is a failure in processing HTTP request or response	Minor
3005	GEN_005	vnfmUnexpectedHttpResponseStatusCodeAlertNotification	Error when there is a unexpected response status code	Minor
3006	GEN_006	vnfmJsonParseErrorAlertNotification	Error when the JSON object parsing fails	Minor
3007	GEN_007	vnfmNoSuchAlgorithmAlertNotification	Error when the requested the algorithm for SSL context is not found	Minor
3008	GEN_008	vnfmKeyManagementAlertNotification	Error if there is a key management issue while initializing	Minor
3009	GEN_009	vnfmTimeoutAlertNotification	Error if the server is taking too long to respond	Minor
3010	GEN_010	vnfmMissingMMIResponseParameterAlertNotification	Error when an expect MMI response parameter is missing	Minor
3011	GEN_011	vnfmInputOutputErrorAlertNotification	An I/O error has occurred	Minor

Table 19-2 (Cont.) General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
3012	GEN_012	vnfmInterruptedErrorAlertNotification	An interrupted error has occurred	Minor
3013	GEN_013	vnfmFileNotFoundAlertNotification	Error if the specified file is not found	Minor
3014	GEN_014	vnfmUnexpectedParseErrorAlertNotification	An unexpected error has occurred while parsing an object or file	Minor
3015	GEN_015	vnfmMissingConfigParamAlertNotification	Error occurred when a configuration file is missing a mandatory parameter	Minor
3016	GEN_016	vnfmUnsupportedConfigParamAlertNotification	Error when a configuration file contains an unsupported parameter	Minor
3017	GEN_017	vnfmValueOutOfBoundsAlertNotification	Error when a value/index is out of range	Minor
3018	GEN_018	vnfmSessionIdErrorAlertNotification	Failed to fetch the session ID	Minor
3019	GEN_019	vnfmIOExceptionErrorAlertNotification	Detected an IOException during processing	Minor
3020	GEN_020	vnfmHttpResourceNotFoundAlertNotification	The requested Http Resource Not Found	Minor
3021	GEN_021	vnfmMMIStatusExceptionErrorAlertNotification	MMI Exception status response	Minor
3022	GEN_022	vnfmNotActiveNodeErrorAlertNotification	Error when the node is not active	Minor
3023	GEN_023	vnfmSoftwareVersionInfoNotFoundAlertNotification	MMI Exception if the Software Version is not found	Minor
3024	GEN_024	vnfmParameterAdditionFailedAlertNotification	MMI Exception if the addition of Parameter failed	Minor
3025	GEN_025	vnfmOperationFailureAlertNotification	Unexpected Operation Failure	Minor
3026	GEN_026	vnfmTemporaryConditionFailureAlertNotification	Temporary Condition Failure	Minor
3027	GEN_027	vnfmJaxbMarshallingErrorAlertNotification	Jaxb Marshalling Error found	Minor
3028	GEN_028	vnfmNoamServerGroupCountErrorAlertNotification	Noam Server group count Error	Minor
3029	GEN_029	vnfmSecureRemoteOperationFailedAlertNotification	Secure Remote Operation Failed	Minor
3030	GEN_030	vnfmXmlParseErrorAlertNotification	XML Parse Error	Minor
3031	GEN_031	vnfmXmlXPathExpressionFailureAlertNotification	The XPath Expression Failed	Minor
3032	GEN_032	vnfmXmlTransformXmlToStringFailureAlertNotification	Converting DOM Xml to String Failure	Minor
3033	GEN_033	vnfmXmiIpAddressIsNotAssignedAlertNotification	Xmi Ip address not assigned to device	Minor
3034	GEN_034	vnfmLoadConfigOperationFailedErrorAlertNotification	Load config operation failure	Minor

Table 19-2 (Cont.) General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
3035	GEN_035	vnfmFileSystemEntityActionFailureAlertNotification	File system entity action failure	Minor
3036	GEN_036	vnfmServerNotReachableAlertNotification	Server not accessible	Minor
3037	GEN_037	vnfmUnsupportedDsrReleaseVersionAlertNotification	DSR Release Invalid	Minor
3038	GEN_038	vnfmCannotDetermineDefaultValueAlertNotification	Default value cannot be determined	Minor
3039	GEN_039	vnfmDsrImagesNotConfiguredAlertNotification	DSR images are not configured for the release	Minor
3040	GEN_040	vnfmDsrImageNotConfiguredVmTypeAlertNotification	DSR image for VM type not configured	Minor
3041	GEN_041	vnfmBulkConfigXmlCreationFailureAlertNotification	Bulk Config XML creation failure	Minor
3046	GEN_046	vnfmUnsupportedVnfTypeAlertNotification	Unsupported VNFM type	Minor
3049	GEN_049	vnfmFileCreationFailureAlertNotification	File creation failed	Minor
3050	GEN_050	vnfmValueNotConfiguredInPropertyFileAlertNotification	Value not configured in property file	Minor
3051	GEN_051	vnfmHeatTemplateStackObjectInstantiationFailureAlertNotification	HeatTemplateStack instantiation failure	Minor
3052	GEN_052	vnfmConfigurationExceptionAlertNotification	Exception while initializing configuration exception	Minor
3053	GEN_053	vnfmWatchDogTimerExceptionAlertNotification	Failed to create Watch Dog Timer	Minor
3054	GEN_054	vnfmInvalidOpenStackResourceAlertNotification	Openstack resource id is not valid	Minor
3055	GEN_055	vnfmUnsupportedFlavorIdAlertNotification	Unsupported VNFM type.	Minor
3056	GEN_056	vnfmReadVnfInstanceAlertNotification	Incorrect VNF Instance Id	Minor
3057	GEN_057	vnfmIllegalInstantiationLevelAlertNotification	Incorrect VNF Instance Id	Minor
3058	GEN_058	vnfmFileNotFoundExceptionAlertNotification	Incorrect VNFM persistent file	Minor
3059	GEN_059	vnfmInvalidFileAlertNotification	Invalid file Error	Minor
3060	GEN_060	vnfmScaledConfigXmlCreationFailureAlertNotification	Bulk Config XML creation failure	Minor
3061	GEN_061	vnfmReadVnfLcmOperationExceptionAlertNotification	Incorrect VNF LCM Operation Id	Minor
3062	GEN_062	vnfmInvalidInstanceNameAlertNotification	vnfInstance Name is already in use	Minor
3063	GEN_063	vnfmInvalidNetworkAlertNotification	Invalid network name.	Minor
3064	GEN_064	vnfmUnsupportedEncodingAlertNotification	Unsupported Encoding Found	Minor

Table 19-2 (Cont.) General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
3065	GEN_065	vnfmReachedMaxAllowedServersPerSignalingVnfAlertNotification	Reached Max allowed servers per signaling VNF	Minor
3066	GEN_066	vnfmReachedMaxAllowedIpfePerSignalingVnfAlertNotification	Reached Max allowed IPFE servers per signaling VNF	Minor
3067	GEN_067	vnfmTerminationFailureAlertNotification	Failed Terminating Stack	Major
3068	GEN_068	vnfmInvalidResourceIdAlertNotification	Exception for invalid resource id	Minor
3069	GEN_069	vnfmRetrieveBulkXmlPersistentFailureAlertNotification	Retrieval of bulk xml from persistent storage failed.	Minor
3070	GEN_070	vnfmRetrievePasswordFailureAlertNotification	Password retrieval failure	Minor
3071	GEN_071	vnfmCloudInitFailureAlertNotification	Cloud Init failed	Major
3073	GEN_073	vnfmInvalidNetworkNameAlertNotification	Network name invalid	Minor
3074	GEN_074	vnfmSSLExceptionAlertNotification	SSL Exception	Minor
3075	GEN_075	vnfmInvalidIPFETargetSetAlertNotification	Invalid IPFE Target Sets for IPFE	Minor
3076	GEN_076	vnfmTsaVipJsonCreationFailureAlertNotification	Tsa Vip json creation failed	Minor
3077	GEN_077	vnfmStateOperationExceptionAlertNotification	VNFM State Operation Exception	Minor
3078	GEN_078	vnfmClientProtocolExceptionAlertNotification	Client Protocol Exception	Minor
3079	GEN_079	vnfmRetrieveLocalIpFailureAlertNotification	Retrieval of Local IP failed	Minor
3080	GEN_080	remoteVnfmChangeStateFailureAlertNotification	Exception occurred while updating the Remote VNFM State Info.	Minor
3081	GEN_081	healVnfFailureAlertNotification	Heal VNF operation Failure.	Major
3082	GEN_082	operateVnfFailureAlertNotification	Operate VNF operation Failure.	Major
3083	GEN_083	modifyVnfFailureAlertNotification	Modify VNF operation Failure.	Major
3084	GEN_084	unknownHostExceptionAlertNotification	Unknown Host Exception while mapping the IP and hostName.	Major
3085	GEN_085	unableToReadAndUpdateFileAlertNotification	Unable to perform file read or update operations.	Major
3086	GEN_086	vnfmReadHostFileAlertNotification	Exception occurred while reading or updating hosts file.	Major
3087	GEN_087	malformedURLExceptionAlertNotification	Exception occurred while Url is not proper.	Major
3088	GEN_088	vnfmIncorrectVnfIdAlertNotification	Exception occurred while vnfId is incorrect.	Major
3089	GEN_089	vnfmVnfNotSupportingMultipleVnfIdAlertNotification	Exception occurred while vnf type is not supporting multiple vnfId.	Major

Table 19-2 (Cont.) General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
3090	GEN_090	vnfmVnfdIdContainsInvalidValueAlertNotification	Exception occurred while vnfdId field contains invalid value.	Major
3091	GEN_091	vnfmCloudInitConfigFileTransferFailedAlertNotification	Exception occurred while cloud config files transfer failed via sftp	Major
3092	GEN_092	vnfmCloudConfigRetryFileTransferFailedAlertNotification	Exception occurred while cloud config file transfer failed via sftp after specific retry	Major
3093	GEN_093	vnfmConvertCsarInstantiateVnfV2ToV1FailureAlertNotification	Exception occurred while converting CSAR instantiate vnf V2 to V1 request Json	Major
3094	GEN_094	sendingNotificationForSubscriptionFailureAlertNotification	Exception occurred while sending notification	Major
3095	GEN_095	vnfmVnfmApiOperationExceptionAlertNotification	VNFM Api Version file reading Operation Exception	Major
3096	GEN_096	vnfmDecryptionFailureExceptionAlertNotification	Exception occurred while decrypting the string	Major
3097	GEN_097	vnfmNoamCleanupTerminateFailureAlertNotification	Exception occurred while DSR / SDS Noam cleanup for GRACEFUL termination of Signaling and DR setups	Major

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.2.4001

Table 19-3 Semantic Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
4001	SEMANTIC_001	vnfmSemanticErrorAlertNotification	Semantic Error Found	Minor
4002	SEMANTIC_002	vnfmInvalidFieldValueParameterAlertNotification	Invalid Field Value Found	Minor
4003	SEMANTIC_003	vnfmInvalidVimConnectionInfoListSizeAlertNotification	Invalid Connection Details in the Vim Connection Information	Minor
4004	SEMANTIC_004	vnfmRequiredParameterMissingAlertNotification	Required Parameters Missing	Minor
4005	SEMANTIC_005	vnfmUnsupportedInputParameterAlertNotification	Unsupported Input Parameters	Minor
4006	SEMANTIC_006	vnfmDatatypeMismatchAlertNotification	Datatype Mismatch Found	Minor
4007	SEMANTIC_007	vnfmValueTooShortParameterAlertNotification	The value of the parameters are too short	Minor
4008	SEMANTIC_008	vnfmValueTooLongAlertNotification	The value of the parameters are too long	Minor
4009	SEMANTIC_009	vnfmIllegalValueAlertNotification	Illegal Value Found	Minor
4010	SEMANTIC_010	vnfmIllegalArgumentAlertNotification	Illegal Argument Found	Minor
4011	SEMANTIC_011	vnfmMissingFixedIpsAlertNotification	Fixed IP addresses are Missing	Minor

Table 19-3 (Cont.) Semantic Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
4012	SEMANTIC_012	vnfmValueLengthMismatchAlertNotification	The length the value has been mismatched	Minor
4013	SEMANTIC_013	vnfmIpNotInRangeAlertNotification	The IP address is out of bounds	Minor
4014	SEMANTIC_014	vnfmInvalidKeyAlertNotification	Invalid Key Found	Minor
4015	SEMANTIC_015	vnfmMismatchedIpVersionAlertNotification	The IP Version has been mismatched	Minor
4016	SEMANTIC_016	vnfmInvalidPasswordAlertNotification	Invalid Password is provided	Minor
4017	SEMANTIC_017	vnfmInvalidSubnetNameAlertNotification	Illegal Value Found	Minor
4018	SEMANTIC_018	vnfmNotSupportedDualIpAlertNotification	Dual Stack not supported	Minor
4019	SEMANTIC_019	vnfmMultipleOccurrenceOfParameterAlertNotification	Multiple occurrence of VIP	Minor
4020	SEMANTIC_020	vnfmInvalidIpfeOptionsAlertNotification	Invalid IPFE Options	Minor
4021	SEMANTIC_021	vnfmInvalidIpfeOptionsForTargetSetsAlertNotification	Invalid IPFE options for Target Sets.	Minor
4022	SEMANTIC_022	vnfmNotSupportedTsaConfigAlertNotification	Invalid Flavor Id for TSA config	Minor
4023	SEMANTIC_023	vnfmInvalidVnfcNamesAlertNotification	Invalid VNFC Names for specific VNF	Minor
4024	SEMANTIC_024	vnfmOperateNoStateChangeAlertNotification	Unable to change instance state while Operate VNF operation	Minor
4025	SEMANTIC_025	vnfmServersNotSupportedAlertNotification	C-level servers not supported for Custom instantiation level id based on flavor	Minor
4026	SEMANTIC_026	vnfmServerCountNotSupportedAlertNotification	C- level server count not supported	Minor
4027	SEMANTIC_027	vnfmInvalidGatewayAlertNotification	Gateway IP is invalid for the subnet	Minor
4028	SEMANTIC_028	vnfmInvalidSegNetworkCombinationAlertNotification	Request body of Signaling VNF does not contains same xmiRepl and imiRepl combination as like NOAM VNF	Minor
4029	SEMANTIC_029	vnfmInvalidSgNameAlertNotification	Request body of Signaling VNF does not contain proper SG name based on IP used in the resource ID field.	Minor
4030	SEMANTIC_030	vnfmInvalidNtplpAlertNotification	Request body contains invalid ntplp.	Minor
	SEMANTIC_031	createForDiscoverOperationFailedv1v2AlertNotification	Incorrect Discover stack Id	Minor

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.3.5001

Table 19-4 OpenStack Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
5001	OPENSTACK_001	vnfmClientCreateFailureAlertNotification	Failed to create Openstack Client.	Minor
5002	OPENSTACK_002	vnfmHeatFileMissingParameterAlertNotification	Mandatory Yaml file for deployment not found	Minor
5003	OPENSTACK_003	vnfmParamMapConvertErrorAlertNotification	Unable to convert parameter Yaml file to parameter list	Minor
5004	OPENSTACK_004	vnfmStackCreateClientErrorAlertNotification	Failed to perform stack create operation due to error on client	Major
5005	OPENSTACK_005	vnfmStackDeleteClientErrorAlertNotification	Failed to delete the stack	Major
5006	OPENSTACK_006	vnfmStackNotFoundErrorAlertNotification	Failed to find the stack by the name	Minor
5007	OPENSTACK_007	vnfmStackCreateServerErrorAlertNotification	Failed to perform stack create operation due to error on server	Major
5008	OPENSTACK_008	vnfmStackGetOutputsTimeoutErrorAlertNotification	Failed to retrieve a stack infrastructure	Minor
5009	OPENSTACK_009	vnfmStackGetOutputsConfigErrorAlertNotification	Failed to open NsaOsProperties file	Minor
5010	OPENSTACK_010	vnfmStackGetOutputsMissingDataErrorAlertNotification	Required data missing from getOutputs response	Minor
5011	OPENSTACK_011	vnfmStackGetOutputsNullValueErrorAlertNotification	Failed to retrieve IPs from stack	Minor
5012	OPENSTACK_012	vnfmInvalidJsonFormatErrorAlertNotification	The generated JSON String has errors	Minor
5013	OPENSTACK_013	vnfmOpenstackCliCommandExecutionFailureAlertNotification	OpenStack command execution failure	Minor
5014	OPENSTACK_014	vnfmStackServiceConfigErrorAlertNotification	Error just before stack creation	Minor
5015	OPENSTACK_015	vnfmStackUpdateClientErrorAlertNotification	Failed to perform stack update operation due to error on client	Major
5016	OPENSTACK_016	vnfmStackUpdateServerErrorAlertNotification	Failed to perform stack update operation due to error on server	Major
5017	OPENSTACK_017	vnfmStackDeleteServerErrorAlertNotification	Failed to perform stack delete operation due to error on server	Major
5018	OPENSTACK_018	vnfmNetworkDetailsNotFoundAlertNotification	Failed to fetch the network details from the provided network	Minor
5019	OPENSTACK_019	vnfmIpDetailsNotFoundAlertNotification	Failed while fetching IP details for the provided resource ID	Minor
5020	OPENSTACK_020	vnfmPortCreationErrorAlertNotification	Failed while creating port from network ID.	Minor
5021	OPENSTACK_021	vnfmNetworkNameFromIdAlertNotification	Failed while fetching network name from network ID.	Minor
5022	OPENSTACK_022	vnfmStackDetailsNotFoundAlertNotification	Failed while fetching stack output from stack.	Minor
5023	OPENSTACK_023	vnfmStackUnknownOperationErrorAlertNotification	Unknown Operation on Open Stack is performed.	Minor
5024	OPENSTACK_024	vnfmOperateVnfStateChangeErrorAlertNotification	vnfmOperateVnfStateChangeErrorAlertNotification	Minor

Table 19-4 (Cont.) OpenStack Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
5025	OPENSTACK_025	vnfmFetchFlavorListErrorAlertNotification	Fetch Openstack Flavor List Failure	Minor
5026	OPENSTACK_026	vnfmFetchImageListErrorAlertNotification	Fetch Openstack Image List Failure	Minor
5027	OPENSTACK_027	vnfmFetchIpFromHostnameAlertNotification	Fetch IP from Hostname Failure	Minor
5028	OPENSTACK_028	vnfmFetchHostnameFromIpAlertNotification	Fetch Hostname from IP Failure	Minor
5029	OPENSTACK_029	vnfmFetchHostnameFromIpAlertNotification	TSA port deletion Failure	Minor
5030	OPENSTACK_030	vnfmNetworkNameFromIdErrorAlertNotification	Failure in fetching Network name From Id	Minor
5031	OPENSTACK_031	vnfmSubnetNameFromIdErrorAlertNotification	Failure in fetching Subnet name from Id	Minor

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.4.6001

Table 19-5 Invalid Gen Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
6001	Invalid_GEN_001	vnfmIncorrectVnfInstanceIdAlertNotification	Incorrect Vnf Instance ID	Minor
6002	Invalid_GEN_002	vnfmIncorrectStackIdOrNameAlertNotification	Incorrect Stack Id or Name	Minor
6003	Invalid_GEN_003	vnfmDiscoverStackIdOrNameAlertNotification	Discovery stack Id or Name already discover by VNFM	Minor
6004	Invalid_GEN_004	vnfmDiscoverStackIdOrNameCreateFailedAlertNotification	Creation of Stack by the provided stack Id or Name failed	Minor
6005	Invalid_GEN_005	vnfmIncorrectVnfLcmOpOccurrenceIdAlertNotification	Incorrect VNF LCM Operation Occurrence Id	Minor
6006	Invalid_GEN_006	vnfmNetworkConfigFileNotExistAlertNotification	Incorrect VNF LCM Operation Occurrence ID.	Minor
6007	Invalid_GEN_007	vnfmIncorrectVnfSubscriptionIdAlertNotification	Incorrect VNF Subscription Operation Occurrence Id	Minor
6008	Invalid_GEN_008	vnfmIncorrectVnfIdAlertNotification	Incorrect VNF Id Operation Occurrence Id	Minor

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.5.7001

Table 19-6 VNFM State Conflict Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
7001	STATE_CONFLICT_001	vnfmVnfAlreadyInstantiatedAlertNotification	The Vnf Instance has already been instantiated	Minor
7002	STATE_CONFLICT_002	vnfmVnfNotInstantiatedAlertNotification	The Vnf Instance has not been instantiated	Minor

Table 19-6 (Cont.) VNFM State Conflict Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
7003	STATE_CONFLICT_003	timeStampTooOldAlertNotification	Request too old	Minor
7004	STATE_CONFLICT_004	vnfcAlreadyInQueueAlertNotification	VNFC already present in priority queue.	Minor

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.6.8001

Table 19-7 VNFM Success Alert

OID	Success Alert ID	Operation	Success Alert Message	Alert Name	Severity
8001	01	STACK CREATE	The vnfm Operation Stack Creation is successful	vnfmStackCreateSuccessAlertNotification	Info
8002	02	STACK UPDATE	The vnfm Operation Stack Update is successful	vnfmStackUpdateSuccessAlertNotification	Info
8003	03	STACK DELETE	The vnfm Operation Stack Terminate is successful	vnfmStackDeleteSuccessAlertNotification	Info
8004	04	STACK Discovery	The vnfm Operation Stack Discover is successful	vnfmStackDiscoverSuccessAlertNotification	Info
8005	05	CLOUD INIT	The vnfm Operation Cloud-Init is successful	vnfmCloudInitSuccessAlertNotification	Info
8006	06	Remote Synchronization	Remote Synchronization is successful.	vnfmRemoteSynchronizationSuccessAlertNotification	Info
8007	07	Change VNFM State	Change VNFM State is successful	changeVnfmStateInfoSuccessAlertNotification	Info
8008	08	MODIFY VNF	Modify VNF is successful	vnfmModifyVnfSuccessAlertNotification	Info
8009	09	OPERATE VNF	The Operate VNF operation is successful.	vnfmOperateVnfSuccessAlertNotification	Info
8010	10	HEAL VNF	The Heal VNF operation is successful.	healVnfSuccessAlertNotification	Info

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.7.2001

OID	Success Alert Message	Alert Name	Severity
2001	One of VNFM Job is Down	vnfmInstanceDownAlertNotification	Critical
2002	Out of Memory	vnfmMemoryUsageAlertNotification	Critical
2003	High CPU Load in the server	vnfmLoadAlertNotification	Critical
2004	Out of disk space	vnfmDiskUsageAlertNotification	Warning

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.8.9001

Table 19-8 VNFM Auth Exception Summary

OID	Success Alert ID	Alert Name	Alert Message	Severity
9001	AUTH_001	vnfmInvalidUserScopeAlertNotification	The Cloud Init is successful	Minor
9002	AUTH_002	vnfmUserAlreadyPresentAlertNotification	User Already Present	Minor
9003	AUTH_003	vnfmInvalidCredentialsEnteredAlertNotification	Invalid username or password entered	Minor
9004	AUTH_004	vnfmSessionExpiredAlertNotification	Session Expired, please login again to continue	Minor
9005	AUTH_005	vnfmInvalidTokenAlertNotification	Invalid Token	Minor
9006	AUTH_006	vnfmNullTokenAlertNotification	Token Field must be present	Minor
9007	AUTH_007	vnfmInvalidStateAlertNotification	VNFM State is invalid	Minor
9008	AUTH_008	vnfmUnauthorizedAccessAlertNotification	Unauthorized Access	Minor

VNFM Constraint Violation Exception Summary

Example OID: .1.3.6.1.4.1.323.5.3.33.1.2.9.1101

Table 19-9 Exception Summary

OID	Success Alert ID	Alert Name	Alert Message	Alert time Stamp	Severity
1101	CONSTRAINT_VIOLATION_001	vnfmConstraintViolationAlertNotification	Constraint Violation in request		Minor
1102	CONSTRAINT_VIOLATION_002	vnfmConstraintNullAlertNotification	Constraint Violation for null for field in request		Minor
1103	CONSTRAINT_VIOLATION_003	vnfmInvalidConstraintSizeAlertNotification	Constraint Violation for size for field in request		Minor
1104	CONSTRAINT_VIOLATION_004	vnfmInvalidConstraintPatternAlertNotification	Constraint Violation for pattern for field in request		Minor

Table 19-10 Exception Summary

OID	Success Alert ID	Alert Name	Alert Message	Alert time Stamp	Severity
1201	AUTH_001	vnfmInvalidGrantAlertNotification	Invalid or Missing value for fetching auth token		Major
1202	AUTH_002	vnfmInvalidRequestAlertNotification	Invalid or Missing value for fetching auth token		Major

Table 19-10 (Cont.) Exception Summary

OID	Success Alert ID	Alert Name	Alert Message	Alert time Stamp	Severity
1203	AUTH_003	vnfmInvalidCredentialsAlertNotification	Invalid or Missing value for fetching auth token		Major

19.2 VNFM MIB File

Oracle VNFM MIB file for exceptions:

```
--
-- VNFM 6.0.2
-- Copyright (C) 2023, Oracle and/or its affiliates. All rights reserved.
--
ORACLEVNFM-MIB DEFINITIONS ::= BEGIN
IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Counter32, TimeTicks,
    Integer32, Unsigned32
                                FROM SNMPv2-SMI
    NOTIFICATION-GROUP          FROM SNMPv2-CONF
    oracleVNFM                   FROM TEKELEC-TOPLEVEL-REG;
oracleVnfmMIB MODULE-IDENTITY
    LAST-UPDATED "201908300000Z"
    ORGANIZATION "Oracle, Inc."
    CONTACT-INFO
        "Tekelec, Inc.
        5200 Paramount Parkway
        Morrisville, NC 27560
        USA

        http://www.oracle.com/support/
        US and Canada: 888.367.8552
        India: +91.124.436.8552
        China: +65.6248.4510
        UK and Europe: +44.1784.467.804"
    DESCRIPTION
        "The MIB module for managing oracleVnfm implementations.
        Copyright (C) Oracle Corp."
--
--
-----
--
--                               REVISION HISTORY
--
-- There should be one REVISION/DESCRIPTION pair for each revision of the
-- file. Revisions should appear in reverse chronological order (the newest
-- revision at the top).
--
-----
    REVISION "201910250000Z"
    DESCRIPTION
```

```

        "Adding VNFM System Alerts."
        ::= { oracleVNFM 1 }

--
-- MIB tables and variables definition
--
oracleVnfmMIBObjects OBJECT IDENTIFIER ::= { oracleVnfmMIB 1 }
oracleVnfmMIBNotifications OBJECT IDENTIFIER ::= { oracleVnfmMIB 2 }
vnfmAlerts OBJECT IDENTIFIER ::= { oracleVnfmMIBObjects 1 }

vnfmExceptionAlertTable OBJECT-TYPE
    SYNTAX SEQUENCE OF VnfmExceptionAlertEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This is the data structure associated to
        exception alerts triggered by the Oracle VNFM."
    ::= { vnfmAlerts 1 }

vnfmExceptionAlertEntry OBJECT-TYPE
    SYNTAX VnfmExceptionAlertEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "This is the data structure associated to
        alerts triggered by Oracle VNFM."
    INDEX { vnfmExceptionAlertID }
    ::= { vnfmExceptionAlertTable 1 }

VnfmExceptionAlertEntry ::=
    SEQUENCE {
        vnfmExceptionAlertID Integer32,
        vnfmExceptionAlertMessage OCTET STRING,
        vnfmExceptionAlertName OCTET STRING,
        vnfmExceptionAlertTimeStamp TimeTicks,
        vnfmExceptionSeverity Integer32
    }

vnfmExceptionAlertID OBJECT-TYPE
    SYNTAX Integer32(0..127)
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The alert ID of the alert being sent; this
        number can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmExceptionAlertEntry 1 }

vnfmExceptionAlertName OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts

```

```

        with raised ones."
    ::= { vnfmExceptionAlertEntry 2 }

vnfmExceptionAlertMessage OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmExceptionAlertEntry 3 }

vnfmExceptionAlertTimeStamp OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The last time any telemetry information was updated."
    ::= { vnfmExceptionAlertEntry 4 }

vnfmExceptionSeverity OBJECT-TYPE
    SYNTAX      INTEGER {
                    critical(3),
                    major(2),
                    minor(1),
                    info(0)}
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmExceptionAlertEntry 5}

vnfmSucessAlertTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VnfmSucessAlertEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This is the data structure associated to
        success alerts triggered by the Oracle VNFM."
    ::= { vnfmAlerts 2 }

vnfmSucessAlertEntry OBJECT-TYPE
    SYNTAX      VnfmSucessAlertEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This is the data structure associated to
        success alerts triggered by Oracle VNFM."
    INDEX { vnfmSuccessAlertID }
    ::= { vnfmSucessAlertTable 1 }

```

```

VnfmSucessAlertEntry ::=
    SEQUENCE {
        vnfmSuccessAlertID      Integer32,
        vnfmOperation           OCTET STRING,
        vnfmSucessAlertMessage  OCTET STRING,
        vnfmSucessAlertTimeStamp TimeTicks,
        vnfmSuccessSeverity     Integer32
    }

vnfmSuccessAlertID OBJECT-TYPE
    SYNTAX      Integer32(0..127)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert ID of the alert being sent; this
        number can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmSucessAlertEntry 1 }

vnfmOperation OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmSucessAlertEntry 2 }

vnfmSucessAlertMessage OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmSucessAlertEntry 3 }

vnfmSucessAlertTimeStamp OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The last time any telemetry information was updated."
    ::= { vnfmSucessAlertEntry 4 }

vnfmSuccessSeverity OBJECT-TYPE
    SYNTAX      INTEGER {
                    crtical(3),
                    major(2),
                    minor(1),
                    info(0)}
    MAX-ACCESS  read-only
    STATUS      current

```

```

DESCRIPTION
    "The alert message of the alert being sent; this
    message can be used to correlate cleared alerts
    with raised ones."
    ::= { vnfmsuccessAlertEntry 5}

vnfmGenExceptionAlertNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { vnfminvalidInstanceNameAlertNotification }
    STATUS        current
    DESCRIPTION
        "The basic notifications implemented by an SNMP entity
        supporting command responder applications."
        ::= { oracleVnfmMIBNotifications 1 }

vnfmSemanticExceptionAlertNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { vnfmunsupportedInputParameterAlertNotification }
    STATUS        current
    DESCRIPTION
        "The basic notifications implemented by an SNMP entity
        supporting command responder applications."
        ::= { oracleVnfmMIBNotifications 2 }

vnfmOpenstackExceptionAlertNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { vnfmunsupportedInputParameterAlertNotification }
    STATUS        current
    DESCRIPTION
        "The basic notifications implemented by an SNMP entity
        supporting command responder applications."
        ::= { oracleVnfmMIBNotifications 3 }

vnfmInvalidGenExceptionAlertNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { vnfmunsupportedInputParameterAlertNotification }
    STATUS        current
    DESCRIPTION
        "The basic notifications implemented by an SNMP entity
        supporting command responder applications."
        ::= { oracleVnfmMIBNotifications 4 }

vnfmStateConflictExceptionAlertNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { vnfmunsupportedInputParameterAlertNotification }
    STATUS        current
    DESCRIPTION
        "The basic notifications implemented by an SNMP entity
        supporting command responder applications."
        ::= { oracleVnfmMIBNotifications 5 }

vnfmsuccessAlertNotificationsGroup NOTIFICATION-GROUP
    NOTIFICATIONS { vnfminvalidInstanceNameAlertNotification }
    STATUS        current
    DESCRIPTION
        "The basic notifications implemented by an SNMP entity
        supporting command responder applications."
        ::= { oracleVnfmMIBNotifications 6 }

vnfmSystemAlertNotificationsGroup NOTIFICATION-GROUP

```

```

NOTIFICATIONS { vnfmInvalidInstanceNameAlertNotification }
STATUS          current
DESCRIPTION
    "The basic notifications implemented by an SNMP entity
    supporting command responder applications."
::= { oracleVnfmMIBNotifications 7 }

vnfmAuthExceptionAlertNotificationsGroup NOTIFICATION-GROUP
NOTIFICATIONS { vnfmUnsupportedInputParameterAlertNotification }
STATUS          current
DESCRIPTION
    "The basic notifications implemented by an SNMP entity
    supporting command responder applications."
::= { oracleVnfmMIBNotifications 8 }

vnfmConstraintViolationExceptionAlertNotificationsGroup NOTIFICATION-GROUP
NOTIFICATIONS { vnfmUnsupportedInputParameterAlertNotification }
STATUS          current
DESCRIPTION
    "The basic notifications implemented by an SNMP entity
    supporting command responder applications."
::= { oracleVnfmMIBNotifications 9 }

vnfmAuthTokenExceptionAlertNotificationsGroup NOTIFICATION-GROUP
NOTIFICATIONS { vnfmUnsupportedInputParameterAlertNotification }
STATUS          current
DESCRIPTION
    "The basic notifications implemented by an SNMP entity
    supporting command responder applications."
::= { oracleVnfmMIBNotifications 10 }

--
-- Start of System Monitoring Alerts
--
vnfmInstanceDownAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmInstanceDownAlertName, vnfmInstanceDownAlertSeverity,
vnfmInstanceDownAlertDescription }
    STATUS          current
    DESCRIPTION
        "Alert for Instance Down."
    ::= { vnfmSystemAlertNotificationsGroup 2001 }

vnfmInstanceDownAlertName OBJECT-TYPE
    SYNTAX          OCTET STRING
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The alert Name of the alert being sent; this
        number can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmInstanceDownAlertNotification 1 }

vnfmInstanceDownAlertSeverity OBJECT-TYPE
    SYNTAX          OCTET STRING
    MAX-ACCESS      read-only
    STATUS          current

```

```

DESCRIPTION
    "The severity of the alert being sent."
    ::= { vnfInstanceDownAlertNotification 2 }

vnfmInstanceDownAlertDescription OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfInstanceDownAlertNotification 3 }

vnfmMemoryUsageAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfMemoryUsageAlertName, vnfMemoryUsageAlertSeverity,
vnfmMemoryUsageAlertDescription }
    STATUS      current
    DESCRIPTION
        "Alert for High Memory Usage."
    ::= { vnfSystemAlertNotificationsGroup 2002 }

vnfmMemoryUsageAlertName OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert Name of the alert being sent; this
        number can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfMemoryUsageAlertNotification 1 }

vnfmMemoryUsageAlertSeverity OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The severity of the alert being sent."
    ::= { vnfMemoryUsageAlertNotification 2 }

vnfmMemoryUsageAlertDescription OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfMemoryUsageAlertNotification 3 }

vnfmLoadAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfLoadAlertName, vnfLoadAlertSeverity,
vnfmLoadAlertDescription }
    STATUS      current
    DESCRIPTION
        "Alert for high Load."

```

```

 ::= { vnfmsystemAlertNotificationsGroup 2003 }

vnfmLoadAlertName OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert Name of the alert being sent; this
        number can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmsystemAlertNotificationsGroup 1 }

vnfmLoadAlertSeverity OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The severity of the alert being sent."
    ::= { vnfmsystemAlertNotificationsGroup 2 }

vnfmLoadAlertDescription OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmsystemAlertNotificationsGroup 3 }

vnfmDiskUsageAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmsystemAlertNotificationsGroup 2004,
              vnfmsystemAlertNotificationsGroup 2005 }
    STATUS      current
    DESCRIPTION
        "Alert for high Disk Usage."
    ::= { vnfmsystemAlertNotificationsGroup 2004 }

vnfmDiskUsageAlertName OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert Name of the alert being sent; this
        number can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmsystemAlertNotificationsGroup 1 }

vnfmDiskUsageAlertSeverity OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The severity of the alert being sent."
    ::= { vnfmsystemAlertNotificationsGroup 2 }

```

```

vnfmDiskUageAlertDescription OBJECT-TYPE
    SYNTAX      OCTET STRING
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The alert message of the alert being sent; this
        message can be used to correlate cleared alerts
        with raised ones."
    ::= { vnfmDiskUageAlertNotification 3 }

--
-- End of System Monitoring Alerts
--

vnfmIllegalArgumentGenAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception for Illegal argument."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3001 }

vnfmNullPointerAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception for a Null Pointer."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3002 }

vnfmWorkingDirectoryErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error while creating the NSA Directory Fails."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3003 }

vnfmHttpClientHandlingErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error when there is a failure in processing HTTP request or
        response."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3004 }

vnfmUnexpectedHttpResponseStatusCodeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current

```

```

DESCRIPTION
    "Error when there is a unexpected response status code."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3005 }

vnfmJsonParseErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error when the JSON object parsing fails."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3006 }

vnfmNoSuchAlgorithmAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error when the requested the algorithm for SSL context is not
found."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3007 }

vnfmKeyManagementAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error if there is a key management issue while initializing."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3008 }

vnfmTimeoutAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error if the server is taking too long to respond."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3009 }

vnfmMissingMMIResponseParameterAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error when an expect MMI response parameter is missing."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3010 }

vnfmInputOutputErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION

```

```

        "An I/O error has occurred."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3011 }

vnfmInterruptedErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "An interrupted error has occurred."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3012 }

vnfmFileNotFoundAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error if the specified file is not found."
        ::= { vnfmGenExceptionAlertNotificationsGroup
3013 }

vnfmUnexpectedParseErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "An unexpected error has occurred while parsing an object or file."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3014 }

vnfmMissingConfigParamAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error occurred when a configuration file is missing a mandatory
parameter."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3015 }

vnfmUnsupportedConfigParamAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error when a configuration file contains an unsupported parameter."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3016 }

vnfmValueOutOfBoundsAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION

```

```

        "Error when a value/index is out of range."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3017 }

vnfmSessionIdErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to fetch the session ID."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3018 }

vnfmIOExceptionErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Detected an IOException during processing."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3019 }

vnfmHttpRequestResourceNotFoundAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "The requested Http Resource Not Found."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3020 }

vnfmMMIStatusExceptionErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "MMI Exception status response."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3021 }

vnfmNotActiveNodeErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error when the node is not active."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3022 }

vnfmSoftwareVersionInfoNotFoundAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "MMI Exception if the Software Version is not found."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3023 }

```

```

vnfmParameterAdditionFailedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "MMI Exception if the addition of Parameter failed."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3024 }

vnfmOperationFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unexpected Operation Failure."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3025 }

vnfmTemporaryConditionFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Temporary Condition Failure."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3026 }

vnfmJaxbMarshallingErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Jaxb Marshalling Error found."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3027 }

vnfmNoamServerGroupCountErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Noam Server group count Error."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3028 }

vnfmSecureRemoteOperationFailedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Secure Remote Operation Failed."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3029 }

vnfmXmlParseErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,

```

```

vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "XML Parse Error."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3030 }

vnfmXmlXPathExpressionFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "The XPath Expression Failed."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3031 }

vnfmXmlTransformXmlToStringFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Converting DOM Xml to String Failure."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3032 }

vnfmXmiIpAddressIsNotAssignedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Xmi Ip address not assigned to device."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3033 }

vnfmLoadConfigOperationFailedErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Load config operation failure."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3034 }

vnfmFileSystemEntityActionFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "File system entity action failure."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3035 }

vnfmServerNotReachableAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }

```

```

STATUS      current
DESCRIPTION
    "Server not accessible."
::= { vnfmGenExceptionAlertNotificationsGroup 3036 }

vnfmUnsupportedDsrReleaseVersionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "DSR Release Invalid."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3037 }

vnfmCannotDetermineDefaultValueAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Default value cannot be determined."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3038 }

vnfmDsrImagesNotConfiguredAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "DSR images are not configured for the release."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3039 }

vnfmDsrImageNotConfiguredVmTypeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "DSR image for VM type not configured."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3040 }

vnfmBulkConfigXmlCreationFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Bulk Config XML creation failure."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3041 }

vnfmUnsupportedVnfTypeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION

```

```

        "Unsupported VNFM type."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3046 }

vnfmFileCreationFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "File creation failed."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3049 }

vnfmValueNotConfiguredInPropertyFileAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Value not configured in property file."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3050 }

vnfmHeatTemplateStackObjectInstantiationFailureAlertNotification NOTIFICATION-
TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "HeatTemplateStack instantiation failure."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3051 }

vnfmConfigurationExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception while initializing configuration exception."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3052 }

vnfmWatchDogTimerExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to create Watch Dog Timer."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3053 }

vnfmInvalidOpenStackResourceAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Openstack resource id is not valid."

```

```

 ::= { vnfmGenExceptionAlertNotificationsGroup 3054 }

vnfmUnsupportedFlavorIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Unsupported VNFM type."
 ::= { vnfmGenExceptionAlertNotificationsGroup 3055 }

vnfmReadVnfInstanceAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Incorrect VNF Instance Id."
 ::= { vnfmGenExceptionAlertNotificationsGroup 3056 }

vnfmIllegalInstantiationLevelAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Incorrect VNF Instance Id."
 ::= { vnfmGenExceptionAlertNotificationsGroup 3057 }

vnfmFileNotFoundExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Incorrect VNFM persistent file."
 ::= { vnfmGenExceptionAlertNotificationsGroup 3058 }

vnfmInvalidFileAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Invalid file Error."
 ::= { vnfmGenExceptionAlertNotificationsGroup 3059 }

vnfmScaledConfigXmlCreationFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Bulk Config XML creation failure."
 ::= { vnfmGenExceptionAlertNotificationsGroup 3060 }

```

```

vnfmReadVnfLcmOperationExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Incorrect VNF LCM Operation Id."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3061 }

vnfmInvalidInstanceNameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "vnfInstance Name is already in use."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3062 }

vnfmInvalidNetworkAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Invalid network name."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3063 }

vnfmUnsupportedEncodingAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Unsupported Encoding Found."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3064 }

vnfmReachedMaxAllowedServersPerSignalingVnfAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Reached Max allowed servers per signaling VNF."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3065 }

vnfmReachedMaxAllowedIpfePerSignalingVnfAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Reached Max allowed IPFE servers per signaling VNF."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3066 }

vnfmTerminationFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,

```

```

vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed Terminating Stack."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3067 }

vnfmInvalidResourceIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed Terminating Stack."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3068 }

vnfmRetrieveBulkXmlPersistentFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed Terminating Stack."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3069 }

vnfmRetrievePasswordFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unable to retrieve password."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3070 }

vnfmCloudInitFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Cloud Init failed"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3071 }

vnfmInvalidNetworkNameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Network name invalid."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3073 }

vnfmSslExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current

```

```

DESCRIPTION
    "SSL Exception."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3074 }

vnfmInvalidIPFETargetSetAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid IPFE Target Sets for IPFE"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3075 }

vnfmTsaVipJsonCreationFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid IPFE Target Sets for IPFE"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3076 }

vnfmStateOperationExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "VNFM State Operation Exception."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3077 }

vnfmClientProtocolExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Client Protocol Exception."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3078 }

vnfmRetrieveLocalIpFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Retrieve Local IP Failure Exception."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3079 }

remoteVnfmChangeStateFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION

```

```

        "Remote VNFM Change State Exception."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3080 }

healVnfFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Heal VNF operation Failure."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3081 }

operateVnfFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Operate VNF operation Failure."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3082 }

modifyVnfFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Modify VNF operation Failure."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3083 }

unknownHostExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unknown Host Exception while mapping the IP and hostName."
        ::= { vnfmGenExceptionAlertNotificationsGroup 3084 }

unableToReadAndUpdateFileAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unable to perform file read and/or update operations"
        ::= { vnfmGenExceptionAlertNotificationsGroup 3085 }

vnfmReadHostFileAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while reading/updating hosts file"
        ::= { vnfmGenExceptionAlertNotificationsGroup 3086 }

```

```

malformedURLExceptionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while Url is not proper"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3087 }

vnfmIncorrectVnfdIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while vndId is incorrect"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3088 }

vnfmVnfNotSupportingMultipleVnfdIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while vnf type not supporting multiple vnfdId"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3089 }

vnfmVnfdIdContainsInvalidValueAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while vnfdId field contains invalid value"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3090 }

vnfmCloudInitConfigFileTransferFailedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while cloud config files transfer failed via
sftp."
    ::= { vnfmGenExceptionAlertNotificationsGroup 3091 }

vnfmCloudConfigRetryFileTransferFailedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Exception occurred while cloud config file transfer failed via sftp
after specific retry"
    ::= { vnfmGenExceptionAlertNotificationsGroup 3092 }

```

```

vnfmSemanticErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Sematic Error Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4001 }

vnfmInvalidFieldValueParameterAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid Field Value Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4002 }

vnfmInvalidVimConnectionInfoListSizeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid Connection Details in the Vim Connection Information."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4003 }

vnfmRequiredParameterMissingAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Required Parameters Missing."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4004 }

vnfmUnsupportedInputParameterAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Unsupported Input Paramters."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4005 }

vnfmDatatypeMismatchAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Datatype Mismatch Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4006 }

vnfmValueTooShortParameterAlertNotification NOTIFICATION-TYPE

```

```

        OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
        STATUS          current
        DESCRIPTION
            "The value of the parameters are too short."
        ::= { vnfmSemanticExceptionAlertNotificationsGroup 4007 }

vnfmValueTooLongAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "The value of the parameters are too long."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4008 }

vnfmIllegalValueAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Illegal Value Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4009 }

vnfmIllegalArgumentAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Illegal Argument Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4010 }

vnfmMissingFixedIpsAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Fixed Ips Missing."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4011 }

vnfmValueLengthMismatchAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "The length the value has been mismatched."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4012 }

vnfmIpNotInRangeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,

```

```

vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "The Ip is out of bounds."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4013 }

vnfmInvalidKeyAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid Key Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4014 }

vnfmMismatchedIpVersionAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "The IP Version has been mismatched."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4015 }

vnfmInvalidPasswordAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid Password is provided."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4016 }

vnfmInvalidSubnetNameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid Subnet Name Found."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4017 }

vnfmNotSupportedDualIpAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Dual Ip Not Supported Exception."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4018 }

vnfmMultipleOccurenceOfParameterAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current

```

```

DESCRIPTION
    "Multiple occurrence of VIP"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4019 }

vnfmInvalidIpfeOptionsAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid IPFE Options"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4020 }

vnfmInvalidIpfeOptionsFortargetSetsAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid IPFE Options for Target Sets"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4021 }

vnfmNotSupportedTsaConfigAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid Flavor Id for tsa config"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4022 }

vnfmInvalidVnfcNamesAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid VNFC Names for specific VNF"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4023 }

vnfmoperateNoStateChangeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unable to change instance state while Operate VNF operation"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4024 }

vnfmServersNotSupportedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "C-level servers not supported for Custom instantiation level id

```

```

based on flavor"
 ::= { vnfmSemanticExceptionAlertNotificationsGroup 4025 }

vnfmServerCountNotSupportedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "C- level server count not supported."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4026 }

vnfmInvalidGatewayAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Gateway IP is invalid for the subnet"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4027 }

vnfmInvalidSegNetworkCombinationAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Request body of Signaling VNF does not contains same xmiRepl and
imiRepl combination as like NOAM VNF"
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4028 }

vnfmInvalidSgNameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Request body of Signaling VNF does not contain proper SG name based
on IP used in resource Id field."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup 4029 }

vnfmInvalidNtpIpAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Request body contains invalid ntpIp."
    ::= { vnfmSemanticExceptionAlertNotificationsGroup
4030 }

vnfmClientCreateFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current

```

```

DESCRIPTION
    "Failed to create Openstack Client."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5001 }

vnfmHeatFileMissingParameterAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Mandatory Yaml file for deployment not found."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5002 }

vnfmParamMapConvertErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unable to convert parameter Yaml file to parameter list."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5003 }

vnfmStackCreateClientErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to perform stack create operation due to error on client."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5004 }

vnfmStackDeleteClientErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to delete the stack."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5005 }

vnfmStackNotFoundErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to find the stack by the name."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5006 }

vnfmStackCreateServerErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to perform stack create operation due to error on server."

```

```

 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5007 }

vnfmStackGetOutputsTimeoutErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Failed to retrieve a stack infrastructure."
 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5008 }

vnfmStackGetOutputsConfigErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Failed to open NsaOsProperties file."
 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5009 }

vnfmStackGetOutputsMissingDataErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Required data missing from getOutputs response."
 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5010 }

vnfmStackGetOutputsNullValueErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "Failed to retrieve IPs from stack"
 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5011 }

vnfmInvalidJsonFormatErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "The generated JSON String has errors."
 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5012 }

vnfmOpenstackCliCommandExecutionFailureAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS current
    DESCRIPTION
        "OpenStack command execution failure."
 ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5013 }

```

```

vnfmStackServiceConfigErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Error just before stack creation."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5014 }

vnfmStackUpdateClientErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to perform stack update operation due to error on client."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5015 }

vnfmStackUpdateServerErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to perform stack update operation due to error on server."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5016 }

vnfmStackDeleteServerErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to perform stack delete operation due to error on server."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5017 }

vnfmNetworkDetailsNotFoundAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed to fetch the network details from the provided network."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5018 }

vnfmIpDetailsNotFoundAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed while fetching IP details for the provided resource ID."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5019 }

vnfmPortCreationErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,

```

```

vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed while creating port from network ID."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5020 }

vnfmNetworkNameFromIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed while fetching network name from network ID."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5021 }

vnfmStackDetailsNotFoundAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failed while fetching stack output from stack."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5022 }

vnfmStackUnknownOperationErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unknown Operation on Open Stack is performed."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5023 }

vnfmOperateVnfStateChangeErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Server state change failed in operate VNF."
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5024 }

vnfmFetchFlavorListErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Fecth Openstack Flavor List Failure"
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5025 }

vnfmFetchImageListErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current

```

```

DESCRIPTION
    "Fecth Openstack Image List Failure"
    ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5026 }

vnfmFetchIpFromHostnameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Fecth Ip from Hostname Failure"
        ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5027 }

vnfmFetchHostnameFromIpAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Fecth Hostname from Ip Failure"
        ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5028 }

vnfmPortDeletionErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "TSA port deletion Failure"
        ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5029 }

vnfmNetwrokNameFromIdErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failure in fetching Network name From Id"
        ::= { vnfmOpenstackExceptionAlertNotificationsGroup 5030 }

vnfmSubnetNameFromIdErrorAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Failure in fetching Subnet name from Id"
        ::= { vnfmOpenstackExceptionAlertNotificationsGroup
5031 }

vnfmIncorectVnfInstanceIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION

```

```

        "Incorrect Vnf Instance ID."
        ::= { vnfmInvalidGenExceptionAlertNotificationsGroup 6001 }

vnfmIncorrectrStackIdOrNameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Incorrect Stack Id or Name."
        ::= { vnfmInvalidGenExceptionAlertNotificationsGroup 6002 }

vnfmDiscoverStackIdOrNameAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Discovery stack Id or Name already discover by VNFM."
        ::= { vnfmInvalidGenExceptionAlertNotificationsGroup 6003 }

vnfmDiscoverStackIdOrNameCreateFailedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Creation of Stack by the provided stack Id or Name failed."
        ::= { vnfmInvalidGenExceptionAlertNotificationsGroup 6004 }

vnfmIncorrectVnfLcmOpOddIdAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Incorrect VNF LCM Operation Occurrence Id."
        ::= { vnfmInvalidGenExceptionAlertNotificationsGroup 6005 }

vnfmNetworkConfigFileNotExistsAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Incorrect VNF LCM Operation Occurrence Id."
        ::= { vnfmInvalidGenExceptionAlertNotificationsGroup 6006 }

vnfmVnfAlreadyInstantiatedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "The Vnf Instance has already been instantiated."
        ::= { vnfmStateConflictExceptionAlertNotificationsGroup 7001 }

```

```

vnfmVnfNotInstantiatedAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "The Vnf Instance has not been instantiated."
    ::= { vnfmStateConflictExceptionAlertNotificationsGroup 7002 }

timeStampTooOldAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "The TimeStamp is too Old."
    ::= { vnfmStateConflictExceptionAlertNotificationsGroup 7003 }

vnfcAlreadyInQueueAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "VNFC already present in priority queue."
    ::= { vnfmStateConflictExceptionAlertNotificationsGroup 7004 }

vnfmStackCreateSucessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSucessAlertMessage,
vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Stack creation is successful."
    ::= { vnfmSucessAlertNotificationsGroup 8001 }

vnfmStackUpdateSucessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSucessAlertMessage,
vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Stack update is successful."
    ::= { vnfmSucessAlertNotificationsGroup 8002 }

vnfmStackDeleteSucessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSucessAlertMessage,
vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Stack deletion is successful."
    ::= { vnfmSucessAlertNotificationsGroup 8003 }

vnfmStackDiscoverSucessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSucessAlertMessage,
vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current

```

```

DESCRIPTION
    "The Stack discovery is successful."
    ::= { vnfmSuccessAlertNotificationsGroup 8004 }

vnfmCloudInitSuccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSuccessAlertMessage,
vnfmSuccessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Cloud Init is successful."
        ::= { vnfmSuccessAlertNotificationsGroup 8005 }

vnfmRemoteSynchronizationSuccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSuccessAlertMessage,
vnfmSuccessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "Remote Synchronization is successful."
        ::= { vnfmSuccessAlertNotificationsGroup 8006 }

changeVnfmStateInfoSuccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSuccessAlertMessage,
vnfmSuccessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "Change VNFM State is successful."
        ::= { vnfmSuccessAlertNotificationsGroup 8007 }

vnfmModifyVnfSuccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSuccessAlertMessage,
vnfmSuccessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Modify VNF instance operation is successful."
        ::= { vnfmSuccessAlertNotificationsGroup 8008 }

vnfmOperateVnfSuccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSuccessAlertMessage,
vnfmSuccessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Operate VNF operation is successful."
        ::= { vnfmSuccessAlertNotificationsGroup 8009 }

healVnfSuccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmSuccessAlertID, vnfmOperation, vnfmSuccessAlertMessage,
vnfmSuccessAlertTimeStamp, vnfmSuccessSeverity }
    STATUS      current
    DESCRIPTION
        "The Heal VNF operation is successful."
        ::= { vnfmSuccessAlertNotificationsGroup 8010 }

vnfmInvalidUserScopeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,

```

```

vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Scope not allowed for this user."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9001 }

vnfmUserAlreadyPresentAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "User Already Present."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9002 }

vnfmInvalidCredentialsEnteredAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid username or password entered."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9003 }

vnfmSessionExpiredAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Session Expired, please login again to continue."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9004 }

vnfmInvalidTokenAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Invalid Token."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9005 }

vnfmNullTokenAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current
    DESCRIPTION
        "Token Field must be present."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9006 }

vnfmInvalidStateAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS          current

```

```

DESCRIPTION
    "VNFM State is invalid."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9007 }

vnfmUnauthorizedAccessAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Unauthorized Access."
    ::= { vnfmAuthExceptionAlertNotificationsGroup 9008 }

vnfmConstraintViolationAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Constraint Violation in request."
    ::= { vnfmConstraintViolationExceptionAlertNotificationsGroup 1101 }

vnfmConstraintNullAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Constraint Violation for null for field in request."
    ::= { vnfmConstraintViolationExceptionAlertNotificationsGroup
1102 }

vnfmInvalidConstraintSizeAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Constraint Violation for size for field in request."
    ::= { vnfmConstraintViolationExceptionAlertNotificationsGroup
1103 }

vnfmInvalidConstraintPatternAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Constraint Violation for pattern for field in request."
    ::= { vnfmConstraintViolationExceptionAlertNotificationsGroup 1104 }

vnfmInvalidGrantAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current

```

```
DESCRIPTION
    "Invalid or Missing value for fetching auth token."
    ::= { vnfmAuthTokenExceptionAlertNotificationsGroup 1201 }

vnfmInvalidRequestAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid or Missing value for fetching auth token."
        ::= { vnfmAuthTokenExceptionAlertNotificationsGroup 1202 }

vnfmInvalidCredentialsAlertNotification NOTIFICATION-TYPE
    OBJECTS { vnfmExceptionAlertID, vnfmExceptionAlertName,
vnfmExceptionAlertMessage, vnfmExceptionAlertTimeStamp,
vnfmExceptionSeverity }
    STATUS      current
    DESCRIPTION
        "Invalid or Missing value for fetching auth token."
        ::= { vnfmAuthTokenExceptionAlertNotificationsGroup
1203 }
END
```

Import HTTPS/SSL Certificate into VNFM

① Note

Diameter must be configured for running the traffic.

20.1 Recombine Existing PEM Keys and Certificates into VNFM

If you have an existing private key and certificates for your server's domain in PEM format, combine them into a PKCS keystore, then convert the PKCS keystore into a Java keystore.

Run the following command:

```
cat <midfile.1.cert.pem> <midfile.2.cert.pem> > intermediates.cert.pem
```

Where <midfile.1.cert.pem> and <midfile.2.cert.pem> are the names of intermediate certificate files.

① Note

If you have multiple intermediate certificates, combine them in any order.

- `openssl pkcs12 -export -in <dsrVnfm.pem> -inkey <dsrVnfm.key> -certfile <intermediate.cert.pem> -passin pass:<existingpassword> -passout pass: xxxx -out vnfm_default.p12 -name "<yourDomainName>"`
For example:
`openssl pkcs12 -export -in dsrVnfm.pem -inkey dsrVnfm.key -passin pass: xxxx -passout pass:xxxx -out vnfm_default.p12 -name dsrvnfm`
- `keytool -importkeystore -srckeystore vnfm_default.p12 -srcstorepass xxxx -srcstoretype PKCS12 -destkeystore vnfm_default.jks -deststorepass xxxx -alias dsrVnfm`
For example:
`keytool -importkeystore -srckeystore vnfm_default.p12 -srcstorepass xxxx -srcstoretype PKCS12 -destkeystore vnfm_default.jks -deststorepass xxxx -alias dsrVnfm`

Note

keytool is the java key and certificate management utility provided by Java. It exist in jre/bin/keytool.

Where,

- <dsrVnfm.pem>: The existing signed certificate file that matches your existing private key.
- <dsrVnfm.key>: The existing private key file.
- <intermediate.cert.pem>: The existing intermediate certificates that complete the chain from your certificate to a root CA.
- <yourDomainName>: The complete domain name of your server.
- <existingpassword>: The password that allows access to the existing key file.
- <yourpassword>: The password that allows access to your new keystore. Provide at least six characters.
- destkeystore file name should be same as mention in the command (vnfm_default.jks).
- srcstorepass is the password that is given in first command (-passout pass: xxxx) and it should also be same as mention in the command (xxxx)
- deststorepass is the password that is used to open the certificate file (vnfm_default.jks) and should also be same as mention in the command (xxxx), because the same file name and password is used in Tomcat Apache to access the SSL certificate.

20.2 Copy Created Certificate (vnfm_default.jks) into VNFM

When the vnfm box is installed, a self-signed certificate is created by VNFM and is placed in the /var/vnfm/certificate/vnfm_default.jks directory by default. This certificate is valid for 365 days.

The client must copy the created certificate with the same name as vnfm_default.jks into the /var/vnfm/certificate/ directory and override the existing vnfm_default.jks certificate.

Note

After making the certificate changes, client must restart the apache tomcat server to reflect the updated certificate in VNFM. To restart the apache tomcat server, see [Reboot Tomcat](#).

20.3 VNFM Self Signed Certificate Generation

1. Create a vnfmCert.conf configuration file as shown in the example below (provide your own details in the respective fields):

```
[ req ]
default_bits = 2048
default_md = sha256
```

```
distinguished_name = req_distinguished_name
req_extensions = req_ext
[ req_distinguished_name ]
countryName = Country Name (2-letter code)
stateOrProvinceName = State or Province Name (full name)
localityName = Locality (e.g. city name)
organizationName = Organization (e.g. company name)
commonName = Common Name (your.domain.com)
[ req_ext ]
subjectAltName = @alt_names
[alt_names]
DNS.1 = *.localhost
DNS.2 = 127.0.0.1
DNS.3 = *.oracle.com
DNS.4 = *.oraclecorp.com
```

2. Generate a key pair and a signing request by running the following command:

```
openssl req -new -keyout dsrVnfm.key -out dsrVnfm.csr -newkey rsa:2048 -
config vnfmCert.conf
```

It will request for password to create private key file.

Note

To skip passphrase in private key, add `-nodes ( read: "No DES encryption")` parameter from the command.

Check if CSR contains the SAN by running the following command:

```
openssl req -noout -text -in sslcert.csr | grep DNS
```

3. Generating a self-signed certificate:

Run the following code to generate a temporary certificate, which is acceptable for 365 days:

```
openssl x509 -req -days 365 -in dsrVnfm.csr -signkey dsrVnfm.key -sha256 -out
dsrVnfm.crt -extfile ca.cnf -extensions req_ext
```

Enter pass phrase for dsrVnfm.key: <type pass phrase of private key>

Check if CSR contains the SAN by running the following command:

```
openssl req -noout -text -in sslcert.csr | grep DNS
```

4. Convert the CRT to PEM format:

Use the openssl tool to convert the CRT to a PEM format that is readable by the reporter:

```
openssl x509 -in dsrVnfm.crt -out dsrVnfm.pem -outform PEM
```

5. To convert the PEM-format keys to Java KeyStores:

```
openssl pkcs12 -export -in dsrVnfm.pem -inkey dsrVnfm.key -passin pass:4srVN6M
-passout pass:4srVN6M -out vnfm_default.p12 -name dsrvnfm
```

6. Convert the `vnfm_default.p12` to a Java keystore `vnfm_default.jks`, by running the following command:

```
keytool -importkeystore -srckeystore vnfm_default.p12 -srcstorepass  
4srVN6M -srcstoretype PKCS12 -destkeystore vnfm_default.jks -deststorepass  
4srVN6M -alias dsrVnfm
```

Note

After importing certificate into java keystore, it is a good practice to check if the certificate information is correct or not. Keytool is the java jdk tool, which exists in `jdk/bin`.

```
keytool -list -v -keystore [enter keystore name] -storepass [enter keystore  
password]
```

Run the following command to delete existing alias from the keystore file (optional):

```
keytool -delete -alias <aliasname> -keystore vnfm_default.jks
```

Note

The `vnfm_default.jks` is the ssl certification file which is being used in VNFM https to establish the ssl connection.

While importing certificate into java keystore, provide `-alias dsrVnfm`. If it prompts to override, type YES.

Use the password "xxxx".

Note

Certificate file name (`vnfm_default.jks`) and alias name (`dsrVnfm`) must be the same as mentioned above.

Multiple HTTPS/SSL Certificate Support

VNFM supports multiple SSL Certificate handling automatically. During SSL handshake with a cloud, the certificate is automatically added to the keystore, which is further used to make subsequent openstack calls.

To check if the certificate has been added successfully to the keystore, the following steps can be performed.

1. Run the following command to check if the certificate has been added into the keystore file successfully:

```
keytool -v -list -keystore <NAME OF THE KEYSTORE FILE > -storepass
<PASSWORD OF THE KEYSTORE>
```

For example: `keytool -v -keystore vnfm_default.jks -storepass password`
This command returns all the certificates present inside the keystore and the corresponding information such as the Alias, Date Of Expiry, Public Keys etc.

2. Run the following code to delete certificates manually from the keystore:

```
keytool -delete -keystore <NAME OF THE KEYSTORE FILE> -store pass
<PASSWORD OF THE KEYSTORE> -alias <keyAlias>
```

① Note

- The keystore has a format for saving the certificates/keys, therefore while removing certificates, only the keytool should be used. Manual deletion within the file can lead to keystore corruption. If the user wants to remove all the certificates from the keystore, it should be done properly by removing the all the certificates with the help of the above command. Simply truncating the data may disrupt the format of the encrypted data and therefore can corrupt the keystore.
- If the following message is displayed on swagger after certificate is imported into `vnfm_default.jks`, then follow the procedure [Reboot Tomcat](#):

```
"Reason: sun.security.validator.ValidatorException: PKIX path
building failed:
sun.security.provider.certpath.SunCertPathBuilderException:unable
to find valid certification path to requested target"
```

VNFM supports openstack vim HTTP and HTTPS client both. To add that in the VNFM, perform the following steps:

1. In VNFM deployed system, add the Openstack certificate in `/var/vnfm/certificate` path to support the Openstack HTTPS client.

```
/var/vnfm/certificate/<certificate name>.pem
```

Example:

```
/var/vnfm/certificate/os-client-certificate-keystore.pem
```

Note

The certificate must be in pem format only.

2. To get the Openstack client certificate, run the command:

```
echo -n | openssl s_client -connect <openstack stack ip>:5000 | \sed -ne  
'/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > os-client-certificate-  
keystore.pem
```

Example:

```
echo -n | openssl s_client -connect 10.10.20.137:5000 | \sed -ne '/-BEGIN  
CERTIFICATE-/,/-END CERTIFICATE-/p' > os-client-certificate-keystore.pem
```

```
echo -n | openssl s_client connect mvl-dev1.us.oracle.com:5000 | \sed -ne  
'/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p' > os-client-certificate-  
keystore.pem
```

The https certificate from Openstack is retrieved and saved in `os-client-certificate-keystore.pem` file.

21.1 Configurable Keystore

There is a script named `configKeystore.py`, which is present in the project directory at `configKeystore.py`. We are using this script to update `keystoreFile` and `keystorePass` attributes in the `apache tomcat server.xml` file. The keystore file name and password must be sent as command-line arguments to the script. For example, `python configKeystore.py keystoreFileName keystorePassword`.

This keystore file and password are used for making openstack calls.

22

NOAM IPv6 Migration

Prerequisite: The xmi and imi network should have two subnet network each, where 1st will be on IPv4 subnet and 2nd will be on IPv6 subnet.

For example:

Network name: ext-net

Table 22-1 Subnets

Name	Network Address	IP Version	Gateway IP
ext-net-subnet	10.75.189.128/25	IPv4	10.75.189.129
ext-net-ipv6-subnet	2606:b400:605:b818::/64	IPv6	2606:b400:605:b818:6e41:6aff:fec7:80bf

Note

The VNFM supports dual subnet, in case a subnet migrate is required, then perform the following steps.

Steps to migrate DSR NOAM on IPv6:

1. Create DSR Noam through VNFM. Provide dual subnet network (xmi and imi) to creating the DSR Noam set up. DSR NOAM will be up and running with IPv4 network interface through VNFM and will create the IPv6 IP address in Openstack for both xmi/imi.
2. Add the allowed address for IPv6 manually through Openstack cli command for both active/standby NOAM.

Note

User should have permission to add the allowed address to port through OpenStack CLI.

Run the following command to add the allowed address pair in port:

```
openstack port set --allowed-address ip-address=<vip ipv6 address> <active noam port id>
```

```
openstack port set --allowed-address ip-address=<vip ipv6 address> <standby noam port id>
```

For example:

```
openstack port set --allowed-address ip-  
address=2606:b400:605:b818:6e41:6aff:fec7:80cf a2d4fe19-d5e8-4a18-  
b08c-0057e68d2bde
```

3. Follow the *Dual IP Stack migration* document to add the IPv6 interface for active or standby NOAM xmi, imi, and VIP.
4. While adding IPv6 interface, use the same IPv6 IP address for active/standby xmi and imi which is created through VNFM for DSR NOAM.
 - a. Go to Openstack GUI.
 - b. Navigate to **Network** -> **<network name>** and locate the active/standby and vip port.
 - c. Open the port to obtain the created IPv4 and IPv6 address.

23

Troubleshooting VNFM

23.1 Debug VNFM

To debug issues during VNFM deployment, check the following log files:

- VNFM logs are located in " /var/vnfm/logs/vnfm.log "
- VNFM boot logs are located in "/var/log/messages".
- Tomcat logs are located in " /usr/share/vnfm/apache-tomcat-9.x.x/logs/catalina.out ".
- SNMP notifier logs are located in "/var/vnfm/logs/snmp_notifier.log"
- Alert Manager logs are located in "/var/vnfm/logs/alertmanager.log"
- Prometheus server logs are located in "/var/vnfm/logs/prometheus.log"
- Node Exporter logs are located in "/var/vnfm/logs/node_exporter.log"
- RSync logs are location in "/var/log/rsync.log"
- Reboot logs are location in "/var/vnfm/logs/reboot.log"
- Cloud-init logs of the DSR/SDS VNFs are located in "/var/TKLC/log/cloud-init/cloud-init.log"

23.2 Adding Route for a New VIM

To add route for a new VIM, run the following commands in root user mode:

1. Open `route-network.sh`, and append the new VIM route address to the data list.
For example: `DataList=10.75.167.0/24,10.75.185.0/24`
2. Run the `/route-network.sh`
3. Run the command `nmcli device show` from the output, get the value of "General Connection" for the device which is attached for VIM interface.
4. Run the following command by passing the vim interface name retrieved from previous step:
 - `nmcli con up '<vim interface name>'`
 - example: `nmcli con up 'Wired connection 1'`

23.3 Reboot Tomcat

To reboot Tomcat, perform the following procedure in the `dsrvnfm` user mode:

1. Go the `/` path.
The `./dsrvnfmStartup.py` script is present.

2. Run this command:

```
./dsrvnfmStartup.py <option for tomcat to start,stop,restart> <ipAddress of vnf, Don't use [] incase of IPv6>
```

Examples:

```
./dsrvnfmStartup.py restart 2606:b400:605:b84a:6e41:6aff:fec7:8100
```

```
./dsrvnfmStartup.py start 10.75.189.201
```

23.4 Resolve HA Alarms on VNF through VNFM Deployed Setup

Perform the following steps to resolve the HA alarms:

1. Check the ping request and response packets from Server-A and Server-B for which alarm has been raised, by running the following command:

```
tcpdump -i <ens4 or enp4s0 or could be different name based on interface configuration> -n "host <server-A>-imi or <server-B>-imi and port 17401 and udp
```

For example: `tcpdump -i eth1 -n "host noam00-17badf67-imi or noam01-17badf67-imi and port 17401 and udp"`

2. If ping request or response packets are not coming from any server, then add security group rule ingress (response) or egress (request) to that instance. The following image shows Ingress response:

Figure 23-1 Ingress Response

☐ Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
☐ Ingress	IPv4	UDP	17401	0.0.0.0/0		-

3. Check the ping packets again after adding the rule and ensure that imi request and response packets are received from each servers, by running the following command:

```
tcpdump -i <ens4 or enp4s0 or could be different name based on interface configuration> -n "<server-A>-imi or <server-B>-imi and port 17401 and udp
```

4. Now restart the cmha process on the node where the alarms are present, by running the following command:

```
pm.set off cmha && sleep 5 && pm.set on cmha
```

Note

If the Node is HA Active, then restarting cmha will cause switch over.

23.5 Adding a Port in Openstack Security Groups

The Security Group Rules define the traffic that is allowed through instances assigned to the security group.

Following are the steps to allow traffic through ports other than the default ports added by VNFM:

1. Open **Security Groups** tab on the Openstack Horizon.
A list of available **Security Groups** appear.
2. From the list, click **Manage Rules** for the required **Security Group**.
3. Select **Add Rule**, provide all the required details in the dialog box.

Note

In the CIDR field, the values for zero address are:

- For IPv4 - 0.0.0.0/0
- For IPv6 - ::/0

4. Click **Add Rule**.

23.6 Debug SNMP System Alerts

Steps to debug SNMP system alerts:

- Check the log files for any errors. For information about list of log files, see [How to debug VNFM](#).
- If default configurations needs to be changed, perform [SNMP System Traps Configurations](#).

23.7 Configure Flavor and Instantiation Levels in VNFM

Steps to configure Flavor and Instantiation Levels in VNFM:

The number of VMs to be allocated to each VNF Flavor and Instantiation Levels are present in the file: `/usr/share/vnfm/openstack/VnfSizing.yaml`

A sample of the file is provided below:

```
dsrSignaling:
  small:
    diameter:
      damp: 2
      ipfe: 2
      stp: 0
      sbr: 0
      udr: 0
  large:
    diameter:
```

```
damp: 32
ipfe: 4
stp: 0
sbr: 0
udr: 0
```

In order to change the default configurations:

- In 'dsrvnfm' user mode, edit this file: /usr/share/vnfm/openstack/VnfSizing.yaml
- Change the number of VMs under the required **VNF Type** → **Instantiation Level Id** → **Flavor Id** and save the file.

For example: In DSR Signaling, under Diameter Flavor Id, small Instantiation Level Id, the user needs 16 DAMPs, 4 IPFEs, the sample of file would be as below:

Edited Sample File

```
dsrSignaling:
  small:
    diameter:
      damp: 16
      ipfe: 4
      stp: 0
      sbr: 0
      udr: 0
  large:
    diameter:
      damp: 32
      ipfe: 4
      stp: 0
      sbr: 0
      udr: 0
```

23.8 Configuring DNS IP in VNFM

Perform the following procedure if DNS IP is not configured.

1. Log in to the VNFM GUI.
2. In the /etc/resolv.conf VNFM parameter file, add the nameserver <DNS Ip> DNS IP if it is unavailable.

Example:

- For the IPv6 setup: nameserver 2606:b400:605:b912:200:5eff:fe00:1f7
- For the IPv4 setup: nameserver 10.250.32.10

3. Add the <IP> cloudname controller IP in the IPv4 section of the /etc/hosts file if the DNS IP is not working.

Example: 10.190.219.195 controller

23.9 Disable SSH Weak Key Exchange Algorithms

To display the list of key exchange algorithms, use the following command:

```
sudo sshd -T |grep -i "^kexalgorithms"
```

Output - Example

```
kexalgorithms curve25519-sha256,curve25519-sha256@ libssh.org,ecdh-sha2-  
nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-  
sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-  
hellman-group-exchange-sha1,diffie-hellman-group14-sha256,diffie-hellman-  
group14-sha1,diffie-hellman-group1-sha1
```

The following two algorithms are considered as weak algorithm based on the Nessus scan result and they should be disabled for key exchange:

- diffie-hellman-group-exchange-sha1 - weak key algorithm
- diffie-hellman-group1-sha1 - weak key algorithm

To skip the weak key exchange algorithms, add the below line in the `/etc/ssh/sshd_config` and `/etc/ssh/ssh_config` files and on all the servers:

```
KexAlgorithms diffie-hellman-group-exchange-sha256,ecdh-sha2-nistp384
```

Note

The `diffie-hellman-group-exchange-sha256` and `ecdh-sha2-nistp384` are added as they are not reported as Weak key algorithms. This result is overridden if `ssh_config` and `sshd_config` contains `KexAlgorithms diffie-hellman-group-exchange-sha256,ecdh-sha2-nistp384` line.

Post execution of `ssh_config` and `sshd_config` update, run the following command:

```
sudo sshd -T |grep -i "^kexalgorithms"
```

Output - Example

```
diffie-hellman-group-exchange-sha256,ecdh-sha2-nistp384
```

For GSSAPI Key exchange algorithms, use the following commands to check GSSAPIAuthentication and GSSAPIKeyExchange parameters:

```
sudo sshd -T | grep -i "gssapiauthentication"  
sudo sshd -T | grep -i "gssapikeyexchange"
```

To disable user authentication and key exchange based on GSSAPI, set the parameters `GSSAPIAuthentication` and `GSSAPIKeyExchange` to **no** in the `/etc/ssh/sshd_config` and `/etc/ssh/ssh_config` files, on all the servers.

```
sudo sed -i 's/.*GSSAPIAuthentication.*/#GSSAPIAuthentication no/i' /etc/ssh/ssh_config && sudo sed -i 's/.*GSSAPIAuthentication.*/{x;/.*GSSAPIAuthentication.*/!{x;h;b;};x;d}' /etc/ssh/sshd_config
```

```
sudo sed -i 's/.*GSSAPIAuthentication.*/# GSSAPIAuthentication no/i' /etc/ssh/ssh_config && sudo sed -i 's/.*GSSAPIAuthentication.*/{x;/.*GSSAPIAuthentication.*/!{x;h;b;};x;d}' /etc/ssh/ssh_config
```

Note

If there is a line with the `GSSAPIKexAlgorithms` or `gssapikexalgorithms` parameter (case-insensitive), remove that line.

Verify `GSSAPIAuthentication` and `GSSAPIKeyExchange` parameters:

```
sudo sshd -T | grep -i "gssapiauthentication"
```

```
sudo sshd -T | grep -i "gssapikeyexchange"
```

After making the changes in the `/etc/ssh/sshd_config` and `/etc/ssh/ssh_config` files, validate the `/etc/ssh/sshd_config` file for errors.

```
sudo sshd -T
```

Then, restart the `sshd` service.

```
sudo service sshd restart
```

23.10 Add or Update Certification

Note

It is recommended to get a SSL certificate certified from a trusted authority, as Nessus scan results in SSL Certificate Cannot Be Trusted, SSL Self-Signed Certificate issues.

To add or update certification, perform the following steps:

1. Run step 1 and 2 from [VNFM Self Signed Certificate Generation](#) procedure to produce `dsrVnfm.key` and `dsrVnfm.csr` files.

Note

For information on self signed certificate, refer to [VNFM Self Signed Certificate Generation](#) section.

2. Send these 2 files to a trusted CA authority for CA signature.
3. Perform step 4 from [VNFM Self Signed Certificate Generation](#) procedure using the `dsrVnfm.crt` file shared back by the trusted CA authority.

Note

Run the remaining steps in the [VNFM Self Signed Certificate Generation](#) section to generate the `vnfm_default.jks` file.

4. After creating `vnfm_default.jks` certificate, before copying it into `/var/vnfm/certificate/`, move the created certificate to backup, if it exists in `/var/vnfm/certificate/` path.

23.11 Changing VNF CLI and DSR GUI Log in Credentials

This section provides information to change VNF CLI and DSR GUI log in username and password.

Prerequisites

- The new credentials are supported by the DSR Image.
- The updated username must be a plain text.
- The updated password must be encrypted using the utility function which is `/configPword.py'passwordToBeEncrypted'`.
Following is an example showing the usage of the script:

Figure 23-2 Usage of Script

```
[root@ol8-vnfm-runner-server2 DSR_Sony]# ./configPword.py 'Welcome@123'
b'Welcome@123'
Jek7ztHM3n07z4ATejR6mrYlhjWv8eQt5mJP0Pyl97cjwfuB+Lq+IBQU5JLw3a6HVuIynC46iIuVutwtTQaNRdmKHtsCsvNmJv
bxZK5ZY6aY5nuIwouANGJQ1nEyxa10nomKBm73+0ml+sIVSqsM37cZuVdFUF1Ylh4k1z3wzYX444hPikRkWZbxTRAzFCuQTERu
Z7fCSzY+g/G9QeH55N3nMR4ptbeTgvxLbXL9I1EUyx+vFS42SeDw06+xp10Wb1JswXCg961s0RYNfGxaySHB8jqcZw==
[root@ol8-vnfm-runner-server2 DSR_Sony]#
```

- The encrypted password must be updated in `VnfmProperties.yaml` file.

Procedure

In the 'dsrvnfm' user mode:

- Open `VnfmProperties` file - `/opt/vnfm/config/VnfmProperties.yaml` to edit the properties.
- To change VNF CLI log in username and password, update `admUsrUsername` and `admUsrPword` fields.
- To change the DSR GUI username and password, update `dsrNoamUsername` and `dsrNoamPword` fields.