Oracle® Communications

Virtual Network Functions Manager Installation and User Guide

Release 4.5
F31308-01
July 2020

ORACLE

Oracle Communications Virtual Network Functions Manager Installation and User Guide, Release 4.5

F31308-01

Contents

1 Introduction

References 1-1
Acronyms and definitions 1-1
Terminology 1-2
Limitations 1-2
My Oracle Support 1-3

2 Virtual Network Functions Manager Overview

Advantage of Using VNFM 2-2

3 VNFM Lifecycle Management Interfaces

4 VNFM OpenStack Prerequisites

Updating the IPFE Image and Flavor 4-3
Enabling the Multiqueue Feature on IPFEs 4-4

5 Install and Configure VNFM

Access VNFM Using the REST Interface 5-7
VNFM Redundancy 5-7
Configurable Server Affinity Policy 5-8
VNFc Nomenclature 5-8
Supported VNFs by VNFM 5-9

6 Upgrading VNFM

7 VNFM User Management

Access Control in VNFM 7-1

Login to VNFM 7-1
Register to VNFM 7-2
Query All User Instances 7-3
Change Password of the User 7-4
Provision and Add User to VNFM 7-5

8 VNFM State Information

Change VNFM State 8-1
VNFM State Parameter Description 8-3
Making VNFM as a Standalone VNFM 8-3
Changing the Default Retry Configuration 8-4

9 Deploying VNFs

Create a VNF Instance 9-2
Query VNF Instance 9-6
Query Individual VNF Instance 9-7
Query All VNF Instances 9-8
Deleting a VNF Instance 9-10
Instantiating VNFM Secondary 9-12
Sample Request for DYNAMIC IP Model 9-12
Sample Request for FIXED IP Model 9-14
Dynamic and Fixed IP Deployment Parameter Description 9-16
Instantiating the Network OAM VNF 9-17
Instantiating the DR Network OAM VNF 9-23
Determining the DR NOAM XMI Resource IDs 9-24
Instantiating the Signaling VNF with Multiple XSI (1, 2 & 4 XSI Interface) 9-30
Determine the NOAM XMI Resource IDs 9-33
Signaling VNF with Multiple XSI Support (1, 2 and 4 XSI only) 9-33
Instantiating Multiple Signaling VNFs 9-60
Instantiating the APIGW VNF 9-61
Instantiating the IDIH VNF 9-64
Determining the Signaling IMI Resource ID: 9-65
Instantiating the SDS Network OAM VNF 9-69
Target set Address Configuration 9-74
IPFE Target Sets Configuration 9-75
Sample Request for Signaling Flavor DIAMETER 9-76
Instantiating the SDS DR Network OAM VNF 9-80
Determining the SDS DR NOAM XMI Resource IDs 9-81
Instantiating the SDS Signaling VNF 9-87

Determining the Signaling IMI Resource IDs 9-89
Determining the SDS NOAM XMI Resource IDs 9-89
Instantiating the ATS Master VNF 9-95
Instantiating the ProvGW VNF 9-99
Non-ConfigDrive VNF Instantiation 9-101
Scale VNF to Level (Only Scale Out) 9-102
Scale VNF to Level using InstantiationLevelId 9-103
Scale VNF to Level using ScaleInfo (Arbitrary Size) 9-107

10 VNF Instantiation across Multi Cloud / Multi Tenant

11 Discover Stack

12 Query LCM Operation

Query Individual LCM Operation 12-1
Query All LCM Operation 12-2

13 Terminating a VNF

Forceful Termination 13-1
Graceful Termination 13-2

14 Changing the Default Configurations

Changing Flavor Names 14-1
Changing Image Names 14-1
Changing Availability Zone 14-1
Changing Profile Name 14-2

15 VNFM SNMP ALERTS

VNFM Alarms 15-3
VNFM MIB File 15-10

16 Import HTTPS/SSL Certificate into VNFM

Recombine Existing PEM Keys and Certificates into VNFM 16-1
Copy Created Certificate (vnfm_default.jks) into VNFM 16-2

VNFM Self Signed Certificate Generation 16-3

17 Multiple HTTPS/SSL Certificate Support

Configurable Keystore 17-1

18 NOAM IPv6 Migration

19 Troubleshooting VNFM

Debug VNFM 19-1
Enable VNFM Logs with Different Log Levels (DEBUG, TRACE, WARN, ERROR) 19-1
Adding Route for a New VIM 19-1
Reboot Tomcat 19-1
Resolve HA Alarms on VNF through VNFM Deployed Setup 19-2
Adding a Port in Openstack Security Groups 19-2
Debug SNMP System Alerts 19-3
Configure Flavor and Instantiation Levels in VNFM 19-3

What's New in This Release

The following new features are introduced in Virtual Network Functions Manager Installation and Upgrade Guide 4.5:

- Support for Target set Address
- Support for Geo Redundancy
- VNFM to support NTP on IPv6 network and multiple NTP resources
- Support for OpenStack Rocky version

List of Figures

2-1	ETSI MANO Specification	2-1
9-1	VNF Create Instance Request	9-3
9-2	Query VNF Instance	9-6
9-3	Deleting a VNF Instance Resource	9-11
9-4	VNF Instantiate Request	9-31
9-5	VNF Create Instance Request	9-88
9-6	VNF Scaling	9-102
12-1	VNF LCM Operation	12-1
13-1	Forceful Termination	13-2
13-2	Graceful Termination	13-3
19-1	Ingress Response	19-2

List of Tables

1-1	Acronyms and definitions	1-1
1-2	Terminologies and Definitions	1-2
4-1	Specific Flavors and respective VNFM Types	4-1
4-2	Openstack Vim Connection Information	4-3
5-1	Parameters and Definitions for VNFM Installation	5-3
5-2	IP Version Mapping	5-6
5-3	Supported VNFs and VMs	5-9
8-1	VNFM State Parameter Description	8-3
9-1	Supported VNFM Network Interfaces	9-1
9-2	Parameters and Definitions for VNF Instance	9-6
9-3	Dynamic IP Deployment Parameters	9-16
9-4	Fixed IP Deployment Parameters	9-17
9-5	Parameters and Definitions for Network OAM VNF	9-23
9-6	Parameters and Definitions for DR Network OAM VNF	9-29
9-7	Supported Instantiation Levels for DSR Signaling VNF	9-31
9-8	Parameters and Definitions for Signaling VNF with Multiple XSI	9-58
9-9	Supported Instantiation levels for DSR APIGW VNF	9-61
9-10	Parameters and Definitions for APIGW VNF	9-63
9-11	Supported Instantiation levels for IDIH VNF	9-65
9-12	Parameters and Definitions for IDIH VNF	9-68
9-13	Parameters and Definitions for SDS Network OAM VNF	9-73
9-14	Parameters and Definitions SDS DR Network OAM VNF	9-86
9-15	SDS Signaling Flavors supported by VNFM	9-89
9-16	Parameters and Definitions for SDS Sigaling VNF	9-94
9-17	Parameters and Definitions for ProvGW VNF	9-101
9-18	Scaling VNF to Level using InstantiationLevelId	9-107
9-19	Parameters and Definitions for Scaling VNF to Level using ScaleInfo	9-112
10-1	Multi cloud/tenant deployment	10-1
13-1	Parameters and Definitions for Terminating VNF	13-1
15-1	General Exception Alert Summary	15-3
15-2	Semantic Exception Alert Summary	15-6
15-3	OpenStack Exception Alert Summary	15-7
15-4	Invalid Gen Exception Alert Summary	15-9
15-5	VNFM State Conflict Exception Alert Summary	15-9
15-6	VNFM Success Alert	15-9

15-7 VNFM Auth Exception Summary 15-10
18-1 Subnets 18-1

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).

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

Acronyms and definitions

An alphabetized list of acronyms used in the document.

Table 1-1 Acronyms and definitions

Acronym	Definition
APIGW	Application Program Interface Gateway
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

Table 1-1 (Cont.) Acronyms and definitions

Acronym	Definition
NOAM	Network Operations and Maintenance
OAM	Operations, Administration, and Maintenance
OHC	Oracle Help Center
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

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.

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.
- Only one HTTPS openstack certificate is supported at any given time.

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

A VNFM automates lifecycle operations for VNFs. Since, each VNF is managed independently, to deploy a DSR it requires creating and instantiating at least two VNFs (one for the network OAM VNF and one for the signaling VNF). Signaling VNFs can be instantiated any time after the network OAM has been instantiated.

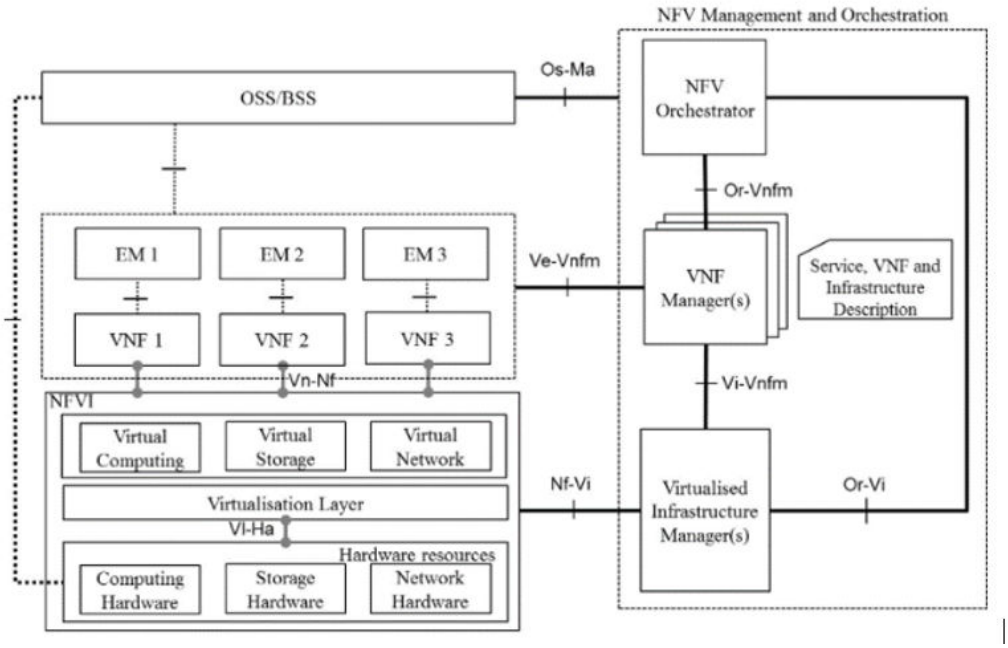
The main objective of the DSR VNFM is to provide an ETSI-compliant VNFM manager. The VNFM would be helpful by:

- 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 VNFM:

Figure 2-1 ETSI MANO Specification



Advantage of Using VNFM

Deployment of Virtual DSR (vDSR) was performed using the following methods that required 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 now deploy a fully operational DSR on OpenStack in less than 15 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.

3
VNFM Lifecycle Management Interfaces

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

- Create 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)
- Terminating VNF
- Discover VNF - Not part of ETSI standard

4
VNFM OpenStack Prerequisites

Following are the prerequisites for using the VNFM:

- An OpenStack instance, ROCKY version. This openstack must be stable for cloud-init completion and proper network connectivity.
- One OpenStack tenant per 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-8.4.0.4.0_87.1.0.vmdk`
  (Optional) Use sections to add and organize related content if another section heading is needed. Where `DSR-8.4.0.4.0_87.1.0.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 Specific Flavors and respective VNFM Types

VNF Type	Image Name	Flavor Name
NOAM, DSR-DBSERVER, DSR-DR-NOAM	`DSR-8.4.0.4.0_87.1.0.vmdk`	`dsr.noam`
SOAM	`DSR-8.4.0.4.0_87.1.0.vmdk`	`dsr.soam`
DA-MP	`DSR-8.4.0.4.0_87.1.0.vmdk`	`dsr.da`
IPFE	`DSR-8.4.0.4.0_87.1.0.vmdk`	`dsr.ipfe`
STP-MP	`DSR-8.4.0.4.0_87.1.0.vmdk`	`dsr.vstp`
SBR	`DSR-8.4.0.4.0_87.1.0.vmdk`	`dsr.sbr`
DSR-APIGWADMIN	`DSRAPIGW-8.4.0.4.0_87.1.0.vmdk`	`dsrapigw.admin`
DSR-APIGWAPP	`DSRAPIGW-8.4.0.4.0_87.1.0.vmdk`	`dsrapigw.app`
UDR	`UDR-12.5.2.0.0_17.14.0.vmdk`	`udr.noam`
DSR-IDIHAPP	`apps-8.2.2.0.0_82.32.0.vmdk`	`appl-idih`

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

VNF Type	Image Name	Flavor Name
DSR-IDIHMEDIATION	mediation-8.2.2.0.0_82.32.0.vmdk	med-idih
DSR-IDIHDB	oracle-8.2.2.0.0_82.32.0.vmdk	db-idih
SDS-NOAM, SDS-QS, SDS-DR-NOAM, SDS-DR-QS	SDS-8.4.0.4.0_87.1.0.vmdk	sds.noam
SDS-SOAM	SDS-8.4.0.4.0_87.1.0.vmdk	sds.dpsoam
SDS-DP	SDS-8.4.0.4.0_87.1.0.vmdk	sds.dp
ATS Master	ATS-8.4.0.4.0-84.13.0.qcow2	ats.master
PROVGW	UDR-PrvGwy-12.6.0.0.0_18.0.0-dev	provGw

VNFC Type	Image Name	Flavor Name	Minimum vCPUs	Minimum RAM (GB)	Minimum Disk (GB)
NOAM, **DSR**-DBSERVER, DSR-DR-NOAM	DSR-8.4.0.4.0_87.1.0.vmdk	dsr.noam	4	6	70
SOAM		dsr.soam	4	6	70
DA-MP		dsr.da	12	16	70
IPFE		dsr.ipfe	6	16	70
STP-MP		dsr.vstp **Note**: Only if using vSTP function	8	6	70
SBR		dsr.sbr	12	25	70
SDS-NOAM, SDS-QS, SDS-DR-NOAM, SDS-DR-QS	SDS-8.4.0.4.0_87.1.0.vmdk	sds.noam	4	32	300
SDS-SOAM		sds.dpsoam	4	12	125
SDS-DP		sds.dp	6	10	125
DSR-**IDIH**APP	apps-8.2.2.0.0_82.32.0.vmdk	appl-idih	4	8	64
DSR-IDIHMEDIATION	mediation-8.2.2.0.0_82.32.0.vmdk	med-idih	8	8	70
DSR-IDIHDB	oracle-8.2.2.0.0_82.32.0.vmdk	db-idih	4	8	200
UDR	UDR-12.5.2.0.0_17.14.0.vmdk	udr.noam	14	64	400

VNFC Type	Image Name	Flavor Name	Minimum vCPUs	Minimum RAM (GB)	Minimum Disk (GB)
DSR-**APIGW**ADMIN	DSRAPIGW-8.4.0.4.0_87.1.0.vmdk	dsrapigw.admin	4	6	70
DSR-APIGWAPP		dsrapigw.app	12	16	70
ATS Master	ats-8.4.0.4.0-84.13.0.qcow2	ats.master	4	16	1024
PROVGW	UDR-PrvGwy-12.6.0.0.0_18.0.0-dev	provGw	4	8	60

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
`id`	Unique Id of the Vim	"vimid"
`vimType`	Virtual Infrastructure Manager (Openstack)	"OpenStack"
`controllerUri`	VIM controller Identity API URI	"https://mvl-dev1.us.oracle.com:5000/v3"
`username`	Username to access openstack controller	"******"
`password`	Password to verified credentials for openstack controller	"******"
`userDomain`	User Domain name for openstack controller	"default"
`projectDomain`	Project Domain Id for openstack controller	"Default"
`tenant`	Tenant name to openstack controller	"VNFM_FT1"

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.

Updating the IPFE Image and Flavor

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

1. Execute the following command to update the multiqueue feature in the IPFE image:

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

2. Execute the following command to update the multiqueue feature in IPFE flavor:

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

Enabling the Multiqueue Feature on IPFEs

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

> **Note:**
>
> This procedure is applicable 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 Updating the IPFE Image and Flavor.

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

5
Install and Configure VNFM

Perform the steps below to install and configure the VNFM:

1. Get one Linux Box which has already installed OpenStack client. If not then install OpenStack client in Linux Box to interact with OpenStack through CLI.
   Steps to install the OpenStack client.
   a. Login as a root user and execute: `yum install python-devel`
   b. Install OpenStack client, by executing: `pip install python-OpenStackclient`
   c. The above command skips importing heatclient plugin, install this plugin by executing:

      ```
      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. Login to OpenStack GUI.
      ii. Go to API Access section tab.
      iii. Click on 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 executing: `source openrc.sh`
      When prompt for password, provide OpenStack controller password.
3. Download the HEAT templates for VNFM installation.

   > **Note:**
   >
   > Download the latest VNFM HEAT templates to your local disk from Oracle Help Center (OHC).

4. Upload the image file to OpenStack:
   a. From the OpenStack GUI, navigate to **Projects > 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 4.5 VM** image. The VNFM Image can be obtained from Oracle Software Delivery Cloud (OSDC) Portal.
   Image name:

   `VNFM_4.5.0.0.0_45.5.0.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/8.4/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
dsrVnfmVmName	Enter a name for the VM. Alphanumeric characters, as well as "-" and "_" are allowed. **Note**: The VM name must not start with "-" and "_".
dsrVnfmImage	Enter the name of the image uploaded in the previous step.
dsrVnfmFlavor	Enter the name of a flavor that is loaded onto OpenStack.

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:
	`"vnfmNetwork": { "network" : "<Network Name>", "subnet" : "<Subnet Name>" }`
	Format for Fixed IP deployment:
	`"vnfmNetwork": { "network" : "<Network Name>", "fixed_ip" : "<ip>" }`

vimNetwork	Enter the name and subnet of a network that VNFM uses to route VIM traffic. (IPv4 or IPv6)
	Format for Dynamic IP deployment:
	`"vimNetwork": { "network" : "<Network Name>", "subnet" : "<Subnet Name>" }`
	Format for Fixed IP deployment:
	`"vimNetwork": { "network" : "<Network Name>", "fixed_ip" : "<ip>" }`

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

Parameter	Value
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.
dsrVnfmAZ	Enter the availability zone to place the VNFM. The "nova" is the default availability zone and is usually the right value.
dsrVnfmVolumeId	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 need 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. Ex1: ntpServer: 10.250.32.10, Ex2: 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 executing:

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

8. To query the VNFM release details after VNFM deployment, execute:`$>./install_vnfm.py --info`

VNFM release information: `Product Name : VNFMProduct Release : 4.5.0`

Refer the following table while choosing the IP versions:

Table 5-2 IP Version Mapping

VNFM External IP Version (REST interface) eth0	VNFM Vim IP Version (VIM interface) eth1	OpenStack Controller VIM IP	DSR IP	Dual Snmp Manager Support	Notes
IPv4	IPv4	IPv4	IPv4	Yes	Supported All the OpenStack traffic/packet will go through VIM IP (eth1) and VNF traffic through default route (eth0).
IPv6	IPv4	IPv4	IPv6	Yes	Supported Default route will add to both interface. Eth0 and Eth1. All the OpenStack traffic/packet will go through VIM IP (eth1) and VNF traffic go through default route (eth0).
IPv6	IPv4	IPv4	IPv4	Yes	Supported Default route will add to both interface, eth0 and eth1. All the OpenStack traffic/packet will go through VIM IP (eth1) and VNF traffic also go through default route IPv4 (eth1) As Vnfm communicates to DSR. IPV6 cannot communicate to IPv4. So, in this case eth1 will communicate to OpenStack and DSR.
IPv6	IPv6	IPv6	IPv6		Not Applicable. The VIM IP version and the controller IP version are different. The communication will never happen. Supported only for IPv6 controller.
IPv6	IPv6	IPv6	IPv6		The MMI call to VNFs fails in case of IPv6. Vms will create but cloud init will fail.
IPv4	IPv6	IPv4			Not Applicable. The VIM IP version and the controller IP version are different. The communication never happens. Supported only for IPv6 controller.

Note:

- **VNFM External IP Version (REST interface) eth0**- Vnfm external IP interface to support the VNFM rest api.
- **VNFM Vim IP Version (VIM interface) eth1**- Vnfm IP that is use to communicate to VIM controller. The vnfm eth1 IP and vim controller IP should be in the same IP version, 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 to 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.

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/).`

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.

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.

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 Polices

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.

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

VNF Instance Name (max 22 Characters)	VNFc Type	Nomenclature (max 5 characters)	Server Name (VM Hostname) (max 30 Characters)
<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>	Prov Gateway	PVGW	<user-input>-PVGW00
<User Input>	DBServer (APIGW)	AGWDB	<user-input>-AGWDB00
<User Input>	dsrApiGwAdmin	AGWAD	<user-input>-AGWAD00
<User Input>	dsrApiGwApp	AGWAP	<user-input>-AGWAP00
<User Input>	DsrIdihApp	IDAPP	<user-input>-IDAPP00
<User Input>	DsrIdihMed	IDMED	<user-input>-IDMED00
<User Input>	DsrIdihDb	IDDB	<user-input>-IDDB00
<User Input>	atsMaster	ATSMA	<user-input>-ATSMA00
<User Input>	atsCore	ATSCO	<user-input>-ATSCO00
<User Input>	atsTools	ATSTO	<user-input>-ATSTO00

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	Supported Dynamic IP VNF	Supported Fixed IP VNF	Supported Dual Stack IP VNF	VNF Dependency	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 NOAM	NOAM (Active/ Standby)	Yes	Yes	Yes		Yes	N/A
DSR DR NOAM	DR NOAM (Active/ Standby)	Yes	Yes	Yes	DSR NOAM	Yes	N/A

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

Supported Dynamic IP VNFs	Supported VNFCs	Supported Dynamic IP VNF	Supported Fixed IP VNF	Supported Dual Stack IP VNF	VNF Dependency	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 Signaling	SOAM (Active/ Standby), DA-MP, STP-MP, IPFE, SBR, UDR	Yes	Yes	Yes (Only for Diameter flavor)	DSR NOAM	Yes*	Yes*
APIGW	DB Server (Active/ Standby), Admin Server, Application Server(s)	Yes		No		No	N/A
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
PROV GW	PROVGW	Yes		No		No	N/A

Yes* -

- Mixed Single Subnet (IPv4 / IPv6 mix) - supported for all flavors.
- Dual subnet and Single subnet mix mode - only DIAMETER flavor supported.

The below table includes the tested combination of DSR-SOAM (Only Diameter Flavor) of XSI's:

VNF TYPE	XSI-1	XS-2	XS-3	XSI-4
DSR-SOAM (Only 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

VNF TYPE	XSI-1	XS-2	XS-3	XSI-4
	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 IPv	Dual Stack IPv4 and IPv6

VNF TYPE	XMI	IMI
SDS - SOAM	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
Upgrading VNFM

The current VNFM stack must be deleted. All the data is stored in the volume that is created during the install procedure. This acts as a persistent storage, so the stack can be safely deleted and the volume is automatically detached from the stack

The user must follow the steps provided in the VNFM Installation procedure with the new IMAGE provided. Flavor, Volume need not be created again. The existing volume ID should 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 required to be attached to other stack is full (around 7GB), then it takes some time to boot the VNFM and load the data.

7

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 **X-Token** for the admin.

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 **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 **X-Token** and use it for other LCM-Operations.

Access Control in VNFM

The `admin` user has access to use all available API's. However, the reader user is restricted to use the following:

- Query Individual VNF Instance
  `https://<VNFM_HOST_IP>:8443/vnflcm/v1/vnf_instances/`
- Query All LCM Operation
  `https://<VNFM_HOST_IP>:8443/vnflcm/v1/vnf_lcm_op_occs`

Login to VNFM

The user must provide the username and the password to generate an authentication token ergo **X-Token**.

Sample Request: Login to VNFM request generated

```
URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnfm_login

Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

Example for Login:

{

"username": "xxxx",
```

```
"password": "xxxx"

}
```

Sample Response: Login to VNFM Response

```
201 Created

Content-Type: application/json

X-Token: Token generated after login

Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnfm_login

{

"tokenId":"eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJRCBvZiB0b2tlbiA6IHRva2VuX1ZORk
0iLCJpYXQiOjE1NzMwMjEyMDYsInN1YiI6IlN1YmplY3Qgb2YgSldUIiwiaXNzIjoiSXNzdWVy
IG9mIFRva2VuOiBPcmFjbGUtRFNSIiwiYXVkIjoiYXVkaWVuY2UgOiBhZG1pbiIsImV4cCI6MT
U3MzAzOTIwNn0.Ep-lKGBZqaO9u_cpj1bSN8DBpWvZoRMQTOYNr18KY8w"

}
```

Where, `username` is the Username of the registered user and `password` is the Password of the registered user.

Register to VNFM

The new user must provide the username, the password & 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/v1/vnfm_register
Accept: application/json

Content-Type: application/json

X-Token: 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

X-Token: Token generated after login

Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnfm_register

{

"response": "Registration Request Sent"

}
```

Where, `username` is the Username of the new user, `password` is the Password of the new user and `access` is the scope of the new user.

Query All User Instances

The admin must provide the **X-Token** 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/v1/view_registration_requests
Accept: X-Token
Content-Type: Text
X-Token: Token generated after login

Example for querying all users:

eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJRCBvZiB0b2tlbiA6IHRva2VuX1ZORk0iLCJpYXQ
iOjE1NzMwMjEyMDYsInN1YiI6IlN1YmplY3Qgb2YgSldUIiwiaXNzIjoiSXNzdWVyIG9mIFR
va2VuOiBPcmFjbGUtRFNSIiwiYXVkIjoiYXVkaWVuY2UgOiBhZG1pbiIsIm4cCI6MTU3MzAz
OTIwNn0.Ep-lKGBZqaO9u_cpj1bSN8DBpWvZoRMQTOYNr18KY8w
```

Sample Response: Querying all user instances Response

```
201 Created
Content-Type: application/json
X-Token: Token generated after login
Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/
view_registration_requests
{
"user": [
{
"username": "xx",
"password": "xx",
"access": "read"
```

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

Where, `X-Token` is the authentication token generated by the admin.

Change Password of the User

The user must provide the username, old password, new password & the X-Token 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/v1/change_password
Accept: application/json
Content-Type: application/json
X-Token: Token generated after login

Example for changing the password for a user:

X-Token:
eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJRCBvZiB0b2tlbiA6IHRva2VuX1ZORk0iLCJpYXQ
iOjE1NzMwMjEyMDYsInN1YiI6IlN1YmplY3Qgb2YgSldUIiwiaXNzIjoiSXNzdWVyIG9mIFR
va2VuOiBPcmFjbGUtRFNSIiwiYXVkIjoiYXVkaWVuY2UgOiBhZG1pbiIsIm4cCI6MTU3MzAz
OTIwNn0.Ep-lKGBZqaO9u_cpj1bSN8DBpWvZoRMQTOYNr18KY8w
{
"username": "xxx",
"newPassword": "xxx",
"oldPassword": "xxx"
}
```

Sample Response: Change Password Response

```
201 Created
Content-Type: application/json
```

```
X-Token: Token generated after login
Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_register/
change_password
{
"response":"Password successfully changed."
}
```

Where, `username` is the username of the existing user, `newPassword` is the new password to be set for the user and `oldPassword` is the existing password of the user.

Provision and Add User to VNFM

The admin must provide the Username, Login credentials and the X-Token to add the credentials in the system.

Sample Request: Provision and Adding the users request generated

```
URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/provision_and_add
Accept: application/json
Content-Type: application/json
X-Token: Token generated after login

Example for provisioning and adding a user:

X-Token :
eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiJJRCBvZiB0b2tlbiA6IHRva2VuX1ZORk0iLCJpYXQ
iOjE1NzMwMjEyMDYsInN1YiI6IlN1YmplY3Qgb2YgSldUIiwiaXNzIjoiSXNzdWVyIG9mIFR
va2VuOiBPcmFjbGUtRFNSIiwiYXVkIjoiYXVkaWVuY2UgOiBhZG1pbiIsIm4cCI6MTU3MzAz
OTIwNn0.Ep-lKGBZqaO9u_cpj1bSN8DBpWvZoRMQTOYNr18KY8w
{
"username": "xxx",
"password": "xxx",
"access": "read/admin"
}
```

Sample Response: Provision and Adding Response

```
201 Created
Content-Type: application/json
X-Token: Token generated after login
Request URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/provision_and_add
{
"response":"Registered successfully."
}
```

Where, `username` is the Username of the user, `password` is the Password of the user and `access` is the scope of the user.

8
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/v1/vnfmState

 X-Token : <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"

}
```

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/v1/vnfmState

Accept: application/json

Content-Type: application/json

X-Token : <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 `TRASIENT` 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.

VNFM State Parameter Description

Table 8-1 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.

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/v1/vnfmState

Accept: application/json

Content-Type: application/json


X-Token : <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"

}
```

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
Deploying VNFs

Prerequisites: A virtual infrastructure satisfying the DSR VNFM OpenStack Prerequisites.

Table 9-1 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
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
APIGW									
APIGWDB	Y	NA	N	N	N	N	N	N	Y
APIGWAdmin	Y	NA	N	N	N	N	N	N	Y
APIGWAPP	Y	NA	N	N	N	N	N	N	Y
ATS									
ATS MASTER	Y	Y(2)	Y	Y	NA	Y	Y	NA	NA
ATS CORE/ TOOLS	Y	NA	Y	NA	NA	NA	N	NA	NA
PROVGW									
PROVGW	Y	NA	N	NA	NA	NA	Y	NA	NA

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 flavor table 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.

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.

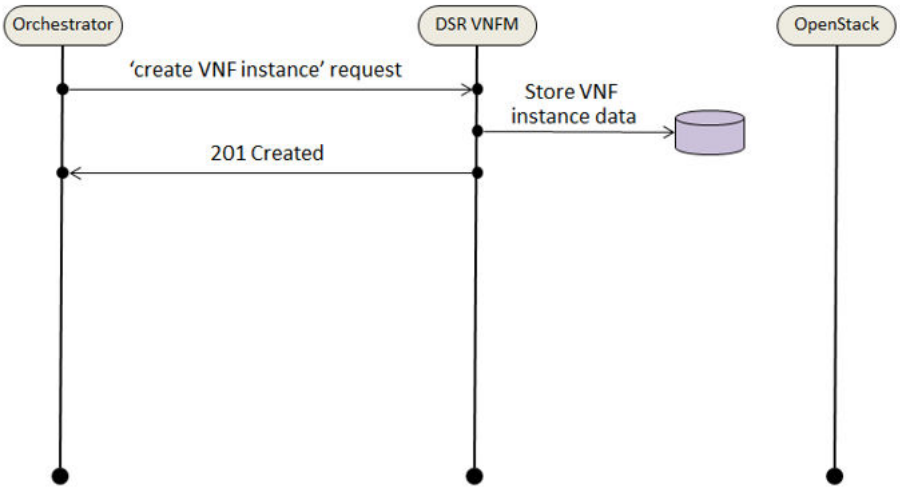
> **Note:**
>
> - Each VNF has its own VNF ID, so if it is required to create a DSR with two signaling VNFs, then issue the request to create a VNF instance three times, once for the network OAM VNF, and once for each signaling VNFs.
> - The `vnfInstanceName` value is defined as per the following:
>   - 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.

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

The following image illustrates the VNF instance creation:

Figure 9-1 VNF Create Instance Request



Sample Request: Create VNF instance request generated.

```
Resource URL: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances
Accept: application/json
Content-Type: application/json
X-Token: Token generated after login
```

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 **APIGW**:

```
{
  "vnfdId": "dsrApiGw",
  "vnfInstanceName": "DemoApiGw",
  "vnfInstanceDescription": "Description for APIGW VNF"
}
```

Example for **IDIH**:

```
{
  "vnfdId": "dsrIdih",
  "vnfInstanceName": "DemoIdih",
  "vnfInstanceDescription": "Description for IDIH VNF"
}
```

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 **ProvGW**:

```
{
  "vnfdId": "provGw",
  "vnfInstanceName": "DemoProvGw",
  "vnfInstanceDescription": "DemoProvGw"
 }
```

Sample Response

`201 Created`

Create VNF Instance Response

`Content-Type: application/json`

`X-Token: Token generated after login`

Resource URL: `https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances`

```
{
   "id":"dsrNetworkOam-b44e9a45-b575-4b30-b580-085d8ddd7015",
   "vnfdId":"dsrNetworkOam",
   "instantiationState":"NOT_INSTANTIATED",
   "vnfInstanceName":"DemoNoam",
   "vnfInstanceDescription":"string",
   "vnfProvider":"Oracle",
   "vnfProductName":"DSR",
   "vnfSoftwareVersion":"DSR_8.4.0.3.0_85.17.0",
   "vnfdVersion":"4.x",
   "onboardedVnfPkgInfoId":"N/A",
   "links":{
      "self":{
         "href":"https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-b44e9a45-b575-4b30-b580-085d8ddd7015"
      },
      "instantiate":{
         "href":"https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-b44e9a45-b575-4b30-b580-085d8ddd7015/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:

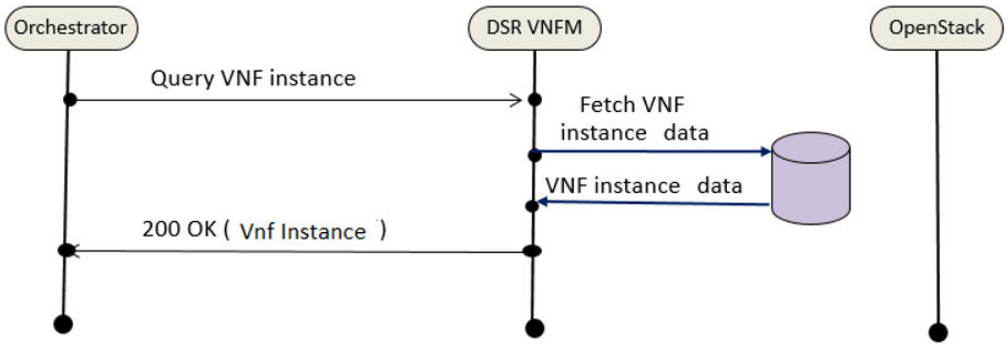
Table 9-2 Parameters and Definitions for VNF Instance

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`	Description of the VNF instance

Query VNF Instance

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

Figure 9-2 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.

Query Individual VNF Instance

Sample Request for Single VNF Instance:

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/<<VNF Instance ID>>

X-Token : <Token generated after login>

Sample Response for Single VNF Instances:

URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/<<VNF Instance ID>>

Accept: application/json

Content-Type: application/json

X-Token: Token generated after login


```
{
   "id": "dsrNetworkOam-793a2420-adab-4347-9667-489ae671b767",
   "vnfdId": "dsrNetworkOam",
   "instantiationState": "NOT_INSTANTIATED",
   "vnfInstanceName": "string",
   "vnfInstanceDescription": "string",
   "vnfProvider": "Oracle",
   "vnfProductName": "DSR",
   "vnfSoftwareVersion": "DSR_8.4.0.3.0_85.17.0",
   "vnfdVersion": "4.x",
   "onboardedVnfPkgInfoId": "N/A",
   "links": {
             "self": {
                      "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/
vnf_instances/dsrNetworkOam-793a2420-adab-4347-9667-489ae671b767"
                     },
             "instantiate": {
                              "href": "https://
<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/dsrNetworkOam-793a2420-
adab-4347-9667-489ae671b767/instantiate"
                             }
           }
 }
Response Body for VNF Instances that are Instantiated
    {
         "id": "dsrNetworkOam-c689e44d-2b93-473f-935a-3bf09957fe9f",
         "vnfdId": "dsrNetworkOam",
         "instantiationState": "INSTANTIATED",
         "vnfInstanceName": "dsrvnfm",
         "vnfInstanceDescription": "dsrvnfm",
         "vnfProvider": "Oracle",
         "vnfProductName": "DSR",
         "vnfSoftwareVersion": "DSR_8.4.0.3.0_85.17.0",
         "vnfdVersion": "4.x",
         "onboardedVnfPkgInfoId": "N/A",
```

```
        "links": {
                    "self": {
                    "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/
vnf_instances/dsrNetworkOam-c689e44d-2b93-473f-935a-3bf09957fe9f"
                },
                "instantiate": {
                    "href": "https://
<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/dsrNetworkOam-
c689e44d-2b93-473f-935a-3bf09957fe9f/instantiate"
                },
                "scaleToLevel": {
                    "href": "https://
<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/dsrNetworkOam-
c689e44d-2b93-473f-935a-3bf09957fe9f/scale_to_level"
               },
                "terminate": {
                    "href": "https://
<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/dsrNetworkOam-
c689e44d-2b93-473f-935a-3bf09957fe9f/terminate"
                }
        },
            "instantiatedVnfInfo": {
            "flavourId": "DSR NOAM",
             "vnfState": "STARTED",
             "extCpInfo": {
                           "id": null,
                           "cpdId": null
                       },
             "scaleStatus": [{
             "aspectId": "NOAM",
             "scaleLevel": "2"
             }]
        },
            "vimConnectionInfo": {
            "id": "vimid",
            "vimType": "OpenStack",
            "interfaceInfo": {
            "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
            "accessInfo": {
                           "username": "dsrat.user",
                           "password": "xxxxxxx",
                           "userDomain": "Default",
                           "projectDomain": "default",
                           "tenant": "DSRAT_Feature_Test1"
                       },
            "extra": {}
    }
 }
```


Query All VNF Instances

Sample Request

URL: `GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances`

Sample Response

URL: `GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances`

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login
```

Response Body for No VNF Instances

```
[]
```

Response Body for all VNF Instances


```
[
  {
    "id": "dsrNetworkOam-38f694dc-be36-4747-814d-5fccd4fa6163",
    "vnfdId": "dsrNetworkOam",
    "instantiationState": "INSTANTIATED",
    "vnfInstanceName": "string",
    "vnfInstanceDescription": "dsrvnfm",
    "vnfProvider": "Oracle",
    "vnfProductName": "DSR",
    "vnfSoftwareVersion": "DSR_8.4.0.3.0_85.17.0",
    "vnfdVersion": "4.x",
    "onboardedVnfPkgInfoId": "N/A",
    "links": {
      "self": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-38f694dc-be36-4747-814d-5fccd4fa6163"
      },
      "instantiate": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-38f694dc-be36-4747-814d-5fccd4fa6163/instantiate"
      },
      "scaleToLevel": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/
dsrNetworkOam-38f694dc-be36-4747-814d-5fccd4fa6163/scale_to_level"
      },
      "terminate": {
        "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/
dsrNetworkOam-38f694dc-be36-4747-814d-5fccd4fa6163/terminate"
      }
    },
    "instantiatedVnfInfo": {
      "flavourId": "DSR NOAM",
      "vnfState": "STARTED",
      "extCpInfo": {
        "id": null,
        "cpdId": null
      },
      "scaleStatus": [
        {
```

```
            "aspectId": "NOAM",
            "scaleLevel": "2"
          }
        ]
      },
      "vimConnectionInfo": {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://dpc1.us.oracle.com:5000/v3"
        },
        "accessInfo": {
          "username": "dsrvnfm",
          "password": "xxxxxxx",
          "userDomain": "Default",
             "projectDomain": "default",
          "tenant": "dsrvnfm"
        },
        "extra": {}
      }
    },
    {
      "id": "dsrNetworkOam-31fd9dc5-bcce-4dfb-ae21-46f07cd3cba5",
      "vnfdId": "dsrNetworkOam",
      "instantiationState": "NOT_INSTANTIATED",
      "vnfInstanceName": "demo",
      "vnfInstanceDescription": "dsrvnfm",
      "vnfProvider": "Oracle",
      "vnfProductName": "DSR",
      "vnfSoftwareVersion": "DSR_8.4.0.3.0_85.17.0",
      "vnfdVersion": "4.2",
      "onboardedVnfPkgInfoId": "N/A",
      "links": {
        "self": {
          "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-31fd9dc5-bcce-4dfb-ae21-46f07cd3cba5"
        },
        "instantiate": {
          "href": "https:// <<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-31fd9dc5-bcce-4dfb-ae21-46f07cd3cba5/instantiate"
        },
        "scaleToLevel": null,
        "terminate": null
      }
    }
  }
```


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.

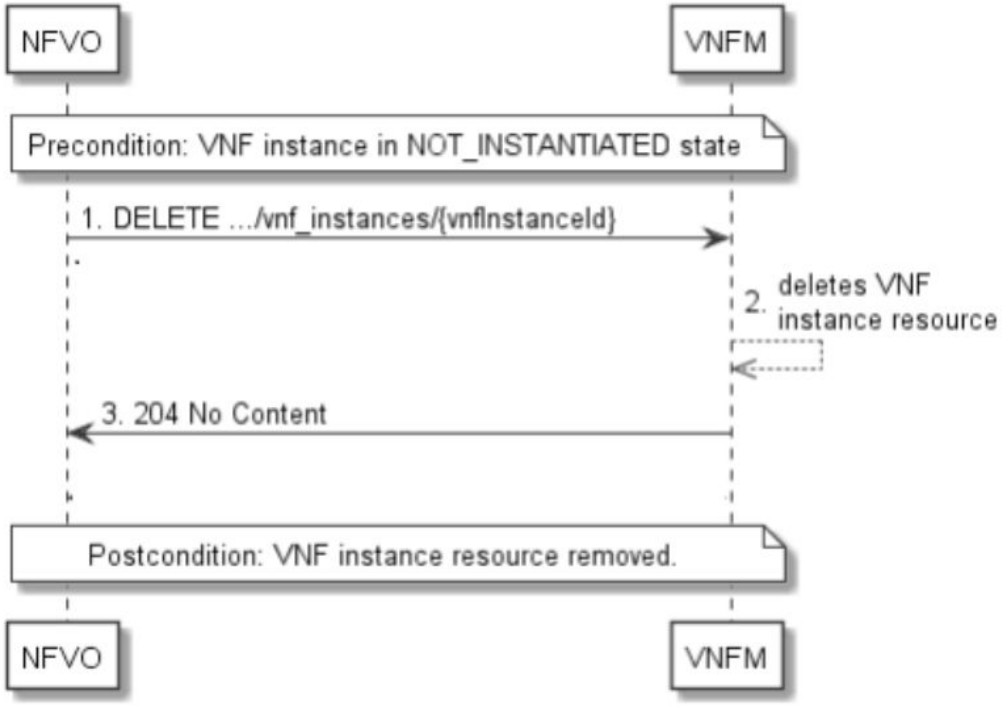
Deletion of a VNF Instance happens in the following sequence:

- 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 9-3 Deleting a VNF Instance Resource



DELETE Operation
`URL: DELETE: https://<<VNFM HOST IP>>:8443/vnfm/v1/vnf_instances/ <<{vnfInstanceId}>>`

DELETE Operation

```
Deletion of a VNF Instance Resource
Response Code : 204

{
     cache-control: private
     expires: Thu, 01 Jan 1970 00:00:00 GMT
     date: Mon, 27 Apr 2020 08:24:09 GMT
     content-type: application/xml
}
```

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 Controller URI
  - User Domain Name
  - Project Domain ID
  - User name
  - Password
  - Tenant name
- 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.

Sample Request for DYNAMIC IP Model

Dynamic deployment model for VNFM Secondary

```
{
```

```
"flavourId": "VNFM SECONDARY",
"instantiationLevelId": "HA",
"extVirtualLinks": "extVirtualLinks",
           "extManagedVirtualLinks": [],
"vimConnectionInfo":[ {
    "id": "vimid",
    "vimType": "OpenStack",
    "interfaceInfo": {
      "controllerUri": "http://oortcloud.us.oracle.com:5000/v3"
    },
    "accessInfo": {
        "username": "dsrvnfm",
        "password": "xxxx",
        "userDomain": "LDAP",
        "projectDomain": "LDAP",
        "tenant": "DSRVNFM_1"
    }
}],
"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": "VNFM_4.5.0.0.0_45.5.0.qcow2",
        "flavor": "vnfm",
        "availabilityZone": "nova"

    }
}
```

Sample Request for FIXED IP Model

Fixed deployment model for VNFM Secondary

```
{
    "flavourId": "VNFM SECONDARY",
    "instantiationLevelId": "HA",
    "extVirtualLinks": "extVirtualLinks",
                "extManagedVirtualLinks": [],
    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://mvl-dev1.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrvnfm",
            "password": "automation",
```

```
                "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_4.5.0.0.0_45.5.0.qcow2",
```

```
        "flavor": "vnfm",
        "availabilityZone": "nova"
    }
}
```

> **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`.

Dynamic and Fixed IP Deployment Parameter Description

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

Table 9-3 Dynamic IP Deployment Parameters

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.

Table 9-3 (Cont.) Dynamic IP Deployment Parameters

Parameter	Description
`availabilityZone (optional)`	Provides the name of logical partitioning in case of host aggregate.

Table 9-4 Fixed IP Deployment Parameters

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.

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

- Tenant name

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

DSR NOAM supports Dual Subnet for XMI and IMI interfaces.

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

```
Accept: application/json
```

```
Content-Type: application/json
```

```
X-Token: Token generated after login
```

Instantiating NOAM Request for dynamic IP deployment (Dual Subnet).

```
{
        "flavourId": "DSR NOAM",
    "instantiationLevelId": "HA",
    "extVirtualLinks": "extVirtualLinks",
                "extManagedVirtualLinks": [],
    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
                  "projectDomain": "default",
            "tenant": "DSR CI"
        }
    }],
    "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",
        "flavor": "dsr.noam",
        "image": "DSR-8.4.0.3.0_85.17.0.vmdk",
        "availabilityZone": "nova",
        "noamAffinityPolicy": "anti-affinity"
    }
}
```

> **Note:**
>
> The "vipSubnetName" field is used only in case of Dual Subnet.

Instantiating NOAM Request for dynamic IP deployment (Single Subnet).

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

    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
             "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
```

```
    "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-8.4.0.3.0_85.17.0.vmdk",
        "availabilityZone": "nova",
        "noamAffinityPolicy": "anti-affinity"
    }
}
```

Instantiating NOAM Request for fixed IP deployment.

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

    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }
    }],
    "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-8.4.0.3.0_85.17.0.vmdk",
        "availabilityZone": "nova",
        "noamAffinityPolicy": "anti-affinity"
    }
}
```

> **Note:**
>
> User must identify available IP addresses to be used in the network. If the user provides an IP address which does not exists in the subnet, the stack creation fails.

Sample Response: Instantiating NOAM Request.

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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>] .
>   For example: `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 9-5 Parameters and Definitions for Network OAM VNF

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
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
imiNetwork	Network used for internal communication of DSR entities
ipVersion	IP version of the network - "IPv4"/"IPv6"
primaryNoamImiIp	IP address for primary NOAM IP of IMI
secondaryNoamImiIp	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
vipSubnetName (In case of Dual Subnet)	Name of VIP subnet to be used only in case of Dual Subnet
noamAffinityPolicy (optional)	openstack affinity policy for NOAM

Instantiating the DR Network OAM VNF

DRNOAM 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 Controller URI
  - User Domain Name
  - Project Domain Id

  - Username
  - Password
  - Tenant 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.

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/v1/vnf_instances/<VNF ID received from create request>/instantiate`

```
Accept: application/json
```

```
Content-Type: application/json
```

```
X-Token: Token generated after login
```

Instantiating DR NOAM Request for Dynamic IP deployment.

```
{
        "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":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
```

```
            "password": "xxxxx",
            "userDomain": "Default",
        "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
    "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-8.4.0.3.0_85.17.0.vmdk",
        "availabilityZone": "nova",
        "drNoamAffinityPolicy":"anti-affinity"
    }
}
```

> **Note:**
>
> The "vipSubnetName" field is used only in case of Dual Subnet.

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":[
      {
         "id":"vimid",
         "vimType":"OpenStack",
         "interfaceInfo":{
            "controllerUri":"https://dpc1.us.oracle.com:5000/v3"
         },
         "accessInfo":{
            "username":"dsrvnfm",
            "password":"xxxx",
            "userDomain":"Default",
            "projectDomain": "default",
            "tenant":"dsrvnfm"
         }
      }
   ],
   "localizationLanguage":"localizationLanguage",
   "additionalParams":{
      "ntpServerIp":"10.250.32.10",
      "xmiNetwork":{
         "name":"ext-net4",
          "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-8.4.0.3.0_85.17.0.vmdk",
   "availabilityZone": "nova",
   "drNoamAffinityPolicy":"anti-affinity"
    }
 }
```

Sample Response: Instantiating DR NOAM Response.

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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**.
- The supported 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 9-6 Parameters and Definitions for DR Network OAM 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 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
drPrimaryNoamIp	IP address of primary DR Noam
drSecondaryNoamIp	IP address of secondary DR Noam
drPrimaryNoamIp	IP address of primary DR Noam
dsPrimaryNoamImiIp	IP address for primary DR NOAM IP of IMI
drSecondaryNoamImiIp	IP address for secondary DR 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
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

Instantiating the Signaling VNF with Multiple XSI (1, 2 & 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 Controller URI
- User Domain Name
- Project Domain Id
- Username
- Password
- Tenant 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.

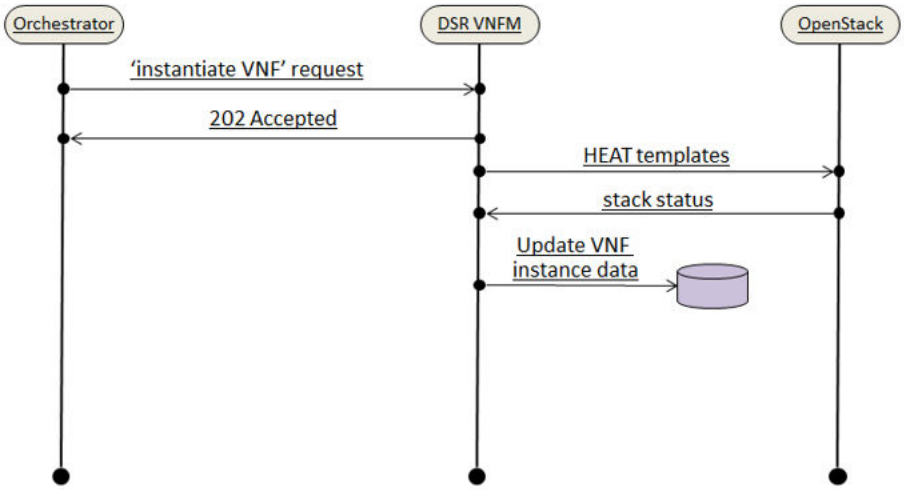
Name of the NOAM SG.

Expected Alarms:

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

The following image illustrates the VNF instantiation:

Figure 9-4 VNF Instantiate Request



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

Table 9-7 Supported Instantiation Levels for DSR Signaling VNF

VNFM Signaling Flavors	Small					Medium					Large				
	DAMP	IPFE	STP	SBR	UDR	DAMP	IPFE	STP	SBR	UDR	DAMP	IPFE	STP	SBR	UDR
DIAMETER	2	2	0	0	0	8	2	0	0	0	32	4	0	0	0
STP	0	0	2	0	0	0	2	8	0	0	0	4	32	0	0
DIAMETER+STP	2	2	2	0	0	8	2	8	0	0	24	4	8	0	0
DIAMETER+SBR	2	2	0	3	0	8	2	0	6	0	32	4	0	9	0

Table 9-7 (Cont.) Supported Instantiation Levels for DSR Signaling VNF

VNFM Signaling Flavors	Small					Medium					Large				
	DAMP	IPFE	STP	SBR	UDR	DAMP	IPFE	STP	SBR	UDR	DAMP	IPFE	STP	SBR	UDR
DIAMETER+STP+SBR	2	2	2	3	0	4	2	4	6	0	24	4	8	9	0
DIAMETER+UDR	2	2	0	0	2	8	2	0	0	2	32	4	0	0	4
STP+UDR	0	0	2	0	2	0	0	8	0	2	0	4	32	0	4
DIAMETER+STP+SBR+UDR	2	2	2	3	2	4	2	4	6	2	16	4	16	9	4

Note:

- In case of UDR flavors, VNFM supports one and two xsi interface.
- 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.

Determine the NOAM XMI Resource IDs

From the OpenStack GUI:

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

> **Note:**
>
> The diagram may take a few minutes to appear.

3. Click on one of the NOAM VMs.
   A screen displays with information about the specific NOAM VM.
4. Save the resource ID for the XMI port provided in the IP addresses section of the screen.

> **Note:**
>
> The IP Addresses section of the popup screen contains information about the network ports and resource IDs assigned to the VM.

5. 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.

> **Note:**
>
> If DSR-NOAM is created on Dual Subnet, then use IPv4 XMI IP's of NOAM while creating SOAM.

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/`).

Signaling VNF with Multiple XSI Support (1, 2 and 4 XSI only)

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

- UDR vnf supports only 1 & 2 xsi interface.

While passing the `xsiNetwork` through request body. Add list of network in the `xsiNetwork`.

For Example

1 xsiNetwork	2 xsiNetwork	4 xsiNetwork

"xsiNetwork": [{
 	
"name":"provider-vlan500",
 	
"subnet":[{
 	
"name": "<subnet-name>",
"ipVersion": "IPv4"
}]
]
	"xsiNetwork": [{
	"name": "provider-vlan500",
	"subnet":[{
	
	"name":"<subnet-name>",
	
	"ipVersion": "IPv4"
	
	}]
	},
	{
	"name": "provider-vlan610",
	
	"subnet":[{
	
	"name":"<subnet-name>",
	
	"ipVersion": "IPv4"
	
	}]
	}
	]
		"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"
		
		}]
		}
		]

The sample request and response provided below represents signaling flavors without SBR such as, DIAMETER, STP & 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: `https://<<myhost-IP>>:8443/vnflcm/v1/vnf_instances/<VNF ID received from create request>/instantiate`

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login
```

Instantiating the first signaling VNF request for Dynamic IP (Dual Subnet) deployment model.

```
{
                "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":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
                "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-8.4.0.3.0_85.17.0.vmdk",
                                "soamAvailabilityZone": "nova",
                                "ipfeFlavor": "dsr.ipfe",
                                "ipfeImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                "ipfeAvailabilityZone": "nova",
                                "daFlavor": "dsr.da",
                                "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
                                "daAvailabilityZone": "nova",
                                "stpFlavor": "dsr.stp",
                                "stpImage":
"DSR-8.4.0.3.0_85.17.0.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.

Instantiating the first signaling VNF request for Dynamic IP deployment model.

```
{
                "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":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
```

```
            "username": "dsrci.user",
            "password": "xxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
                "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-8.4.0.3.0_85.17.0.vmdk",
                                "soamAvailabilityZone": "nova",
                                "ipfeFlavor": "dsr.ipfe",
                                "ipfeImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                "ipfeAvailabilityZone": "nova",
                                "daFlavor": "dsr.da",
                                "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
                                "daAvailabilityZone": "nova",
                                "stpFlavor": "dsr.stp",
                                "stpImage":
```

```
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                    "stpAvailabilityZone": "nova",
                                    "soamAffinityPolicy":"anti-affinity",
                                    "ipfeAffinityPolicy":"anti-affinity",
                                    "daAffinityPolicy":"soft-anti-affinity",
                                    "stpAffinityPolicy":"soft-anti-affinity"
                }
}
```

Instantiating the first signaling VNF request for Fixed IP deployment.

```
{
   "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":[
      {
         "id":"vimid",
         "vimType":"OpenStack",
         "interfaceInfo":{
            "controllerUri":"https://dpc1.us.oracle.com:5000/v3"
         },
         "accessInfo":{
            "username":"dsrvnfm",
            "password":"xxxx",
            "userDomain": "Default",

            "projectDomain": "default",
            "tenant":"dsrvnfm"
         }
      }
   ],
   "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"
                ]
             }
          }]
 },

                                    "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"
                ]
             }
          }]
 },
                     "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",
        "soamFlavor": "dsr.soam",
        "soamImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
        "soamAvailabilityZone": "nova",
        "ipfeFlavor": "dsr.ipfe",
        "ipfeImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
        "ipfeAvailabilityZone": "nova",
        "daFlavor": "dsr.da",
        "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
        "daAvailabilityZone": "nova",
        "stpFlavor": "dsr.stp",
        "stpImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
        "stpAvailabilityZone": "nova",
        "soamAffinityPolicy":"anti-affinity",
        "ipfeAffinityPolicy":"anti-affinity",
        "daAffinityPolicy":"soft-anti-affinity",
        "stpAffinityPolicy":"soft-anti-affinity"
     }
  }
```

Sample Response

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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, DIAMETER+STP+SBR, DIAMETER+STP+SBR+UDR) with multiple xsi (1, 2, or 4 xsi interface) generated for Dynamic IP deployment model.

Resource URL: `https://<<myhost-IP>>:8443/vnflcm/v1/vnf_instances/<VNF ID received from create request>/instantiate`

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

{
                "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":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxx",
            "userDomain": "Default",
```

```
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
                "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":
"dsrNetworkOam_NOAM_32cd6138_SG",
                                   "soamFlavor": "dsr.soam",
                                   "soamImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                   "soamAvailabilityZone": "nova",
                                   "ipfeFlavor": "dsr.ipfe",
                                   "ipfeImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                   "ipfeAvailabilityZone": "nova",
                                   "daFlavor": "dsr.da",
                                   "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
                                   "daAvailabilityZone": "nova",
                                   "sbrFlavor": "dsr.sbr",
                                   "sbrImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                   "sbrAvailabilityZone": "nova",
                                   "sbrAffinityPolicy":"affinity",
                                   "soamAffinityPolicy":"anti-affinity",
                                   "ipfeAffinityPolicy":"anti-affinity",
                                   "daAffinityPolicy":"soft-anti-affinity"
                }
}
```

Instantiating the signaling VNF request with SBR (DIAMETER+SBR, DIAMETER+STP+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":[
 {
 "id":"vimid",
 "vimType":"OpenStack",
 "interfaceInfo":{
 "controllerUri":"https://dpc1.us.oracle.com:5000/v3"
```

```
},
"accessInfo":{
"username":"dsrvnfm",
"password":"xxxx",
"userDomain": "Default",
"projectDomain": "default",
"tenant":"dsrvnfm"
}
}
],
"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-8.4.0.3.0_85.17.0.vmdk",
 "soamAvailabilityZone": "nova",
 "ipfeFlavor": "dsr.ipfe",
 "ipfeImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
 "ipfeAvailabilityZone": "nova",
 "daFlavor": "dsr.da",
 "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
 "daAvailabilityZone": "nova",
 "sbrFlavor": "dsr.sbr",
 "sbrImage": "DSR-8.4.0.3.0_85.17.0.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":[
      {
         "id":"vimid",
         "vimType":"OpenStack",
         "interfaceInfo":{
            "controllerUri":"https://dpc1.us.oracle.com:5000/v3"
```

```
        },
        "accessInfo":{
           "username":"dsrvnfm",
           "password":"xxxx",
           "userDomain": "Default",

           "projectDomain": "default",
           "tenant":"dsrvnfm"
        }
     }
  ],
  "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-8.4.0.3.0_85.17.0.vmdk",
        "soamAvailabilityZone": "nova",
```

```
      "ipfeFlavor": "dsr.ipfe",
      "ipfeImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
      "ipfeAvailabilityZone": "nova",
      "daFlavor": "dsr.da",
      "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
      "daAvailabilityZone": "nova",
      "udrFlavor": "udr.noam",
      "udrImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
      "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/v1/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
```

Sample Request

For signaling flavors with DIAMETER+STP+SBR+UDR with multiple xsi (2 XSI interface) for Fixed IP deployment model.

```
{"flavourId": "DIAMETER+STP+SBR+UDR",
   "instantiationLevelId": "small",
   "extVirtualLinks": "extVirtualLinks",
   "extManagedVirtualLinks": [
     {
       "id": "id1",
       "virtualLinkDescId": "active NOAM",
       "resourceId": "790bf9f7-8834-4c3a-bd17-5544ef5e6848"
     },
     {
       "id": "id2",
       "virtualLinkDescId": "standby NOAM",
       "resourceId": "1776d877-f643-45d6-b6da-bf1a540a01d1"
     }
   ],
   "vimConnectionInfo": [
     {
       "id": "vimid",
       "vimType": "OpenStack",
       "interfaceInfo": {
         "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
```

```
        },
        "accessInfo": {
          "username": "dsrvnfm",
          "password": "xxxxxx",
          "userDomain": "Default",
          "projectDomain": "default",
          "tenant": "dsrvnfm"
        }
      }
    ],
    "localizationLanguage": "localizationLanguage",
   "additionalParams": {
      "xmiNetwork": {
                    "name": "ext-net4",
                    "subnet":[{
                                "name": "ext-net4-subnet",
                                "ipVersion": "IPv4",
                                "fixedIps": {
                                            "primarySoamXmiIp":
"10.75.218.91",

"secondarySoamXmiIp":"10.75.218.223",
                                            "soamVip":"10.75.218.36",

"primaryUdrXmiIp":"10.75.218.180",

"secondaryUdrXmiIp":"10.75.218.205",
                                            "udrVip":"10.75.218.121",
                                            "dampXmiIps":
["10.75.218.242","10.75.218.194"],
                                            "ipfeXmiIps":
["10.75.218.159","10.75.218.198"],
                                            "stpXmiIps":
["10.75.218.241","10.75.218.128"],
                                            "sbrXmiIps":
["10.75.218.147","10.75.218.209","10.75.218.105"]
                                           }
                             }]
                },

    "imiNetwork": {
                    "name": "imi-int",
                    "subnet":[{
                                "name": "imi-int-sub",
                                "ipVersion": "IPv4",
                                "fixedIps": {
                                            "primarySoamImiIp":
"192.167.2.0",

"secondarySoamImiIp":"192.167.2.1",

"primaryUdrImiIp":"192.167.2.2",

"secondaryUdrImiIp":"192.167.2.3",
                                            "dampImiIps":
```

```
["192.167.2.4","192.167.2.5"],
                                                "ipfeImiIps":
["192.167.2.6","192.167.2.7"],
                                                "stpImiIps":
["192.167.2.8","192.167.2.9"],
                                                "sbrImiIps":
["192.167.2.10","192.167.2.11","192.167.2.12"]
                                               }
                              }]
                 },

     "sbrNetwork": {
                     "name": "ext-net4",
                     "subnet":[{
                                 "name": "ext-net4-subnet",
                                 "ipVersion": "IPv4",
                                 "fixedIps": {
                                                "sbrNetworkIps":
["10.75.218.231","10.75.218.236","10.75.218.244"]
                                                }
                               }]
                  },
     "xsiNetwork": [{
                     "name": "ext-net4",
                     "subnet":[{
                                "name": "ext-net4-subnet",
                                "ipVersion": "IPv4",
                                 "fixedIps": {
                                                "dampXsiIps":
["10.75.218.238","10.75.218.47"],
                                                "ipfeXsiIps":
["10.75.218.239","10.75.218.93"],
                                                "stpXsiIps":
["10.75.218.214","10.75.218.19"],
                                                "primaryUdrXsiIps":
["10.75.218.228"],
                                                "secondaryUdrXsiIps":
["10.75.218.235"]
                                               }

                               }]
                              },
                               {
                                 "name": "ext-net4",
                                 "subnet":[{
                                            "name": "ext-net4-subnet",
                                            "ipVersion": "IPv4",
                                            "fixedIps": {
                                                          "dampXsiIps":
["10.75.218.230","10.75.218.225"],
                                                          "ipfeXsiIps":
["10.75.218.49","10.75.218.245"],
                                                          "stpXsiIps":
["10.75.218.170","10.75.218.224"],
```

```
"primaryUdrXsiIps": ["10.75.218.233"],

"secondaryUdrXsiIps":["10.75.218.227"]
                                                    }
                                        }]
                              }
     ],
     "ntpServerIp": "10.250.32.10",
     "primaryNoamVmName": "NOAM00-d8fc80a2",
     "noamSgName": "dsrNetworkOam_NOAM_d8fc80a2_SG"
     "soamFlavor": "dsr.soam",
     "soamImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
     "soamAvailabilityZone": "nova",
     "ipfeFlavor": "dsr.ipfe",
     "ipfeImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
     "ipfeAvailabilityZone": "nova",
     "daFlavor": "dsr.da",
     "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
     "daAvailabilityZone": "nova",
     "udrFlavor": "udr.noam",
     "udrImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
     "udrAvailabilityZone": "nova",
     "sbrFlavor": "dsr.sbr",
     "sbrImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
     "sbrAvailabilityZone": "nova",
     "sbrAffinityPolicy":"affinity",
     "soamAffinityPolicy":"anti-affinity",
     "ipfeAffinityPolicy":"anti-affinity",
     "daAffinityPolicy":"soft-anti-affinity",
     "stpAffinityPolicy":"soft-anti-affinity",
     "udrAffinityPolicy":"anti-affinity"
   }}
```

Sample Response

```
Instantiating the signaling VNF with DIAMETER+STP+SBR+UDR response

202 Accepted

Headers:
location: https:// <<VNFM HOST IP>>:8443/vnflcm/v1/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
```

Sample Request

For signaling flavors with DIAMETER+STP+SBR+UDR with multiple xsi (2 XSI interface) for Dynamic IP deployment model.

```
{"flavourId": "DIAMETER+STP+SBR+UDR",
   "instantiationLevelId": "small",
   "extVirtualLinks": "extVirtualLinks",
```

```
 "extManagedVirtualLinks": [
   {
     "id": "id1",
     "virtualLinkDescId": "active NOAM",
     "resourceId": "790bf9f7-8834-4c3a-bd17-5544ef5e6848"
   },
   {
     "id": "id2",
     "virtualLinkDescId": "standby NOAM",
     "resourceId": "1776d877-f643-45d6-b6da-bf1a540a01d1"
   }
 ],
 "vimConnectionInfo": [
   {
     "id": "vimid",
     "vimType": "OpenStack",
     "interfaceInfo": {
       "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
     },
     "accessInfo": {
       "username": "dsrvnfm",
       "password": "xxxxxx",
       "userDomain": "Default",
       "projectDomain": "default",
       "tenant": "dsrvnfm"
     }
   }
 ],
 "localizationLanguage": "localizationLanguage",
"additionalParams": {
   "xmiNetwork": {
     "name": "ext-net4",
     "subnet":[{
                "name": "ext-net4-subnet",
                "ipVersion": "IPv4"
               }]
   },
   "imiNetwork": {
     "name": "imi-int",
     "subnet":[{
                 "name": "imi-net-sub",
                 "ipVersion": "IPv4"
               }]
   },
   "xsiNetwork": [
     {
       "name": "ext-net4",
      "subnet":[{
                 "name": "ext-net4-subnet",
                 "ipVersion": "IPv4"
                }]
     },
     {
       "name": "ext-net4",
       "subnet":[{
```

```
                "name": "ext-net4-subnet",
                "ipVersion": "IPv4"
               }]
    }
  ],
   "sbrNetwork": {
                "name": "ext-net4",
                "subnet":[{
                          "name": "ext-net4-subnet",
                          "ipVersion": "IPv4"
                         }]
  },
  "ntpServerIp": "10.250.32.10",
  "primaryNoamVmName": "NOAM00-d8fc80a2",
  "noamSgName": "dsrNetworkOam_NOAM_d8fc80a2_SG"
  "soamFlavor": "dsr.soam",
  "soamImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
  "soamAvailabilityZone": "nova",
  "ipfeFlavor": "dsr.ipfe",
  "ipfeImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
  "ipfeAvailabilityZone": "nova",
  "daFlavor": "dsr.da",
  "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
  "daAvailabilityZone": "nova",
  "udrFlavor": "udr.noam",
  "udrImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
  "udrAvailabilityZone": "nova",
  "sbrFlavor": "dsr.sbr",
  "sbrImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
  "sbrAvailabilityZone": "nova",
  "sbrAffinityPolicy":"affinity",
  "soamAffinityPolicy":"anti-affinity",
  "ipfeAffinityPolicy":"anti-affinity",
  "daAffinityPolicy":"soft-anti-affinity",
  "stpAffinityPolicy":"soft-anti-affinity",
  "udrAffinityPolicy":"anti-affinity"
}}
```

Sample Response

```
Instantiating the signaling VNF with DIAMETER+STP+SBR+UDR response
202 Accepted

Headers:
location: https:// <<VNFM HOST IP>>:8443/vnflcm/v1/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:

- 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)`

Detailed explanation of XMI, IMI and XSI Network

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

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 9-8 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
xsiNetwork	Network used for DSR signaling traffic
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

Table 9-8 (Cont.) Parameters and Definitions for Signaling VNF with Multiple XSI

Parameters	Definitions
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)
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

Table 9-8 (Cont.) Parameters and Definitions for Signaling VNF with Multiple XSI

Parameters	Definitions
ipfeImage (optional)	image used for OpenStack deploys
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
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
tsa (optional)	for configuring tsa
optionSets (optional)	for configuring optionSets of tsa

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.

Instantiating the APIGW VNF

To start APIGW deployment, it is required to instantiate an APIGW VNF. Before deploying the VNF, make sure the following information is available:

The VNF ID for a previously created APIGW 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 will carry APIGW traffic.

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

> **Note:**
>
> This should be a different network than the one that carries APIGW 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. Swagger specifications can be found post VNFM installation at (https://<VNFM IP>:8443/docs/vnfm/).

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

Table 9-9 Supported Instantiation levels for DSR APIGW VNF

APIGW Flavors supported by VNFM	Small			Medium			Large		
	ADMIN	APP	DB	ADMIN	APP	DB	ADMIN	APP	DB
APIGW	1	1	Active/ Standby	1	2	Active/ Standby	1	3	Active/ Standby

Sample Request

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

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

Instantiating APIGW Request generated.

{
  "flavourId": "APIGW",
  "instantiationLevelId": "small",
  "extVirtualLinks": "extVirtualLinks",
  "extManagedVirtualLinks": [],
      "vimConnectionInfo": [
    {
      "id": "vimid",
      "vimType": "OpenStack",
      "interfaceInfo": {
        "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "username": "dsrat.user",
        "password": "xxxx",
        "userDomain": "Default",
        "projectDomain": "default",
        "tenant": "DSR AT Dev 2"
      }
    }
  ],
  "localizationLanguage": "localizationLanguage",
  "additionalParams": {
    "ntpServerIp": "10.250.32.10",
    "keyName": "apiGwKey",
    "xmiNetwork": {
      "name": "ext-net3",
      "ipVersion": "IPv4",
                        "xmiSubnetName" : "ext-net3-subnet"
    },
    "imiNetwork": {
      "name": "imi-net",
      "ipVersion": "IPv4",
                        "imiSubnetName" : "imi-subnet"
    },
    "xsiNetwork": {
      "name": "ext-net3",
      "ipVersion": "IPv4",
                        "xsiSubnetName" : "ext-net3-subnet"
    },
    "externalLoadBalancer": "10.10.10.10",
    "mtu":"9000",
    “dsrMPList”: “10.10.10.4:49152”,
    “appServersVolumeIds”: [“320f3557-9a0a-4c13-9d19-d4f0f755b941”]
    "apiGwAppFlavor": "dsrapigw.app",
```

```
    "apiGwAdminFlavor": "dsrapigw.admin",
    "dbServerFlavor": "dsr.noam",
    "apiGwAppImage": "DSRAPIGW-8.4.0.3.0_85.17.0.vmdk",
    "apiGwAdminImage": "DSRAPIGW-8.4.0.3.0_85.17.0.vmdk",
    "dbServerImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
    "apigwAvailabilityZone": "nova"
  }
}
```

Sample Response

Instantiating APIGW Request

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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 6 minutes to become fully operational. Use the DSR GUI to determine when the VNF is operational.

The supported flavor is **APIGW**.

The keyName is the name of the key that generates public & private key in openstack dynamically while creating stack and this key is used to communicate over admin to app server & DB server.

One push script executes and enables the OCSG. After successful execution of one push script, the Admin portal and the App portals GUI comes up.

APIGW is configured automatically and it does not require manual intervention.

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

Table 9-10 Parameters and Definitions for APIGW 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.

Table 9-10 (Cont.) Parameters and Definitions for APIGW VNF

Parameters	Definitions
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.
xsiNetwork	Network used for DSR signaling traffic
ntpServerIp	IP of the NTP server
keyName	Name of key-pair to be generated
externalLoadBalancer	The external load balancer IP where the API is exposed on
mtu	Maximum transfer Unit to do scp file. For different cloud values will be different. (Ex: For oort and mvl-dev1 mtu value will be 9000 and for dpc1 it will be 1500.)
dsrMPList	List of DSR MPs
appServersVolumeIds	A JSON Array containing the volume IDs of the volumes created by the user that is mounted to the individual App Servers. The size/length of this array should be equal to the number of App Servers, which in turn depends on the flavor chosen by the user.
apiGwAdminFlavor (optional)	flavor used for openstack deploys
apiGwAppFlavor (optional)	flavor used for openstack deploys
dberverFlavor (optional)	flavor used for openstack deploys
apiGwAdminImage (optional)	image used for openstack deploys
apiGwAppImage (optional)	image used for openstack deploys
dberverImage (optional)	image used for openstack deploys
apigwAvailabilityZone (optional)	name of logical partitioning in case of host aggregate

Instantiating the IDIH VNF

To start IDIH deployment, it is required to instantiate a signaling VNF. Before deploying the VNF, make sure 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 will carry 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 private network in the selected OpenStack instance that will carry OAM traffic. A signaling stack must be brought up first and then 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.

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

Determining the Signaling IMI Resource ID:

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.

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

Table 9-11 Supported Instantiation levels for IDIH VNF

IDIH Flavors supported by VNFM	APP (Small)	MEDIATION (Small)	DB (Small)
IDIH	1	1	1

Sample Request

Instantiating IDIH Request for dynamic IP deployment

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

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

{
   "flavourId":"IDIH",
   "instantiationLevelId":"small",
   "extVirtualLinks":"extVirtualLinks",
   "extManagedVirtualLinks":[
 {
         "id":"id1",
```

```
         "virtualLinkDescId":" Network ID of the network used for intra-
site communication(imi) with Signalling VNF",
         "resourceId":"aae72b3d-d189-4464-a217-58bb0320065b"
 }
   ],
   "vimConnectionInfo":[
      {
         "id":"vimid",
         "vimType":"OpenStack",
         "interfaceInfo":{
            "controllerUri":"https://oortcloud.us.oracle.com:5000/v3"
         },
         "accessInfo":{
            "username":"dsrat.user",
            "password":"xxxx",
            "userDomain":"Default",
            "projectDomain": "default",
            "tenant":"DSRAT_Feature_Test4"
         }
      }
   ],
   "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-8.2.2.0.0_82.30.0.vmdk",
       "idihMedImage": "mediation-8.2.2.0.0_82.30.0.vmdk",
  "idihDbImage": "oracle-8.2.2.0.0_82.30.0.vmdk",
  "idihAvailabilityZone": "nova"
      }
}
```

Instantiating IDIH Request for fixed IP deployment

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

   "extManagedVirtualLinks":[
 {
         "id":"id1",
         "virtualLinkDescId":" Network ID of the network used for intra-
```

```
site communication(imi) with Signalling VNF",
         "resourceId":"aae72b3d-d189-4464-a217-58bb0320065b"
 }
   ],
   "vimConnectionInfo":[
      {
         "id":"vimid",
         "vimType":"OpenStack",
         "interfaceInfo":{
            "controllerUri":"https://oortcloud.us.oracle.com:5000/v3"
         },
         "accessInfo":{
            "username":"dsrat.user",
            "password":"xxxx",
            "userDomain":"Default",
            "projectDomain": "default",
            "tenant":"DSRAT_Feature_Test4"
         }
      }
   ],
   "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-8.2.2.0.0_82.30.0.vmdk",
         "idihMedImage": "mediation-8.2.2.0.0_82.30.0.vmdk",
         "idihDbImage": "oracle-8.2.2.0.0_82.30.0.vmdk",
         "idihAvailabilityZone": "nova"
      }
 }
```

Sample Response

Instantiating IDIH Request

```
202 Accepted
Headers:
{
```

```
     location: https://localhost:8443/vnflcm/v1/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 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.

The supported flavor is IDIH.

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

Table 9-12 Parameters and Definitions for IDIH 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 Private network (imi) of the Signaling VNF
xmiNetwork	Network that is used to provide access to the DSR entities (GUI, ssh), and for inter-site communication
IdihIntNetwork	Private network for communication between application, mediation and database servers
ntpServerIp	IP of the NTP server
idihDbXmiIp	Fixed IP address of IDIH database server
idihMedXmiIp	Fixed IP address of IDIH mediation server
idihAppXmiIp	Fixed IP address of IDIH application server
idihAppImage (optional)	image used for openstack deploys
idihMedimage (optional)	image used for openstack deploys
idihDbImage (optional)	image used for openstack deploys
idihAppFlavor (optional)	flavor used for openstack deploys
idihMedFlavor (optional)	flavor used for openstack deploys
idihDbFlavor (optional)	flavor used for openstack deploys
idihAvailabilityZone (optional)	name of logical partitioning in case of host aggregate

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

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: Sample Request for DYNAMIC IP deployment model

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

```
Accept: application/json
```

```
Content-Type: application/json
```

```
X-Token: Token generated after login

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

    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
    "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-8.4.0.3.0_85.17.0.vmdk",
        "sdsQsImage": "SDS-8.4.0.3.0_85.17.0.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

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

    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
    "localizationLanguage": "localizationLanguage",
    "additionalParams": {
        "xmiNetwork": {
            "name": "ext-net8",
            "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-8.4.0.3.0_85.17.0.vmdk",
        "sdsQsImage": "SDS-8.4.0.3.0_85.17.0.vmdk",
        "sdsNoamAvailabilityZone": "nova",
        "sdsQsAvailabilityZone": "nova",
        "sdsNoamAffinityPolicy":"anti-affinity"
    }
}
```

Sample Response

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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. Once 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 9-13 Parameters and Definitions for SDS Network OAM VNF

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
`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
`sdsQsAvailabilityZone (optional)`	name of logical partitioning in case of host aggregate

Table 9-13 (Cont.) Parameters and Definitions for SDS Network OAM VNF

Parameter	Definitions
`sdsNoamAffinityPolicy (optional)`	openstack affinity policy for SDS NOAM

Target set Address Configuration

Target set Address (TSA) is used for load balancing the traffic, so that IPFE routes all the incoming traffic and in return traffic is directly routed through gateway instead of IPFE.

For configuring TSA, configure the following on active Soam(Soam Vip):

- IPFE optionSets
- IPFE target Sets

> **Note:**
>
> Use DSR-8.4.0.6.0-89.1.1 or later version to enable this feature.

IPFE Option Sets Configuration

These are the parameters for configuring OptionSets: `IpfeA1IpAddress`, `IpfeA2IpAddress`, `IpfeB1IpAddress`, and `IpfeB2IpAddress`. These options accept the IMI IP address of IPFE servers. These are logical names that facilitates binding with IPFE servers.

The `OptionSets` parameter in the request body is not mandatory. So, by default, VNFM configures `OptionSets` in the following ways:

- 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 pairs 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 pairs 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)

To change the default configurations, send `OptionSets` in request body and configurations are done accordingly.

IPFE Target Sets Configuration

Target sets can be configured for a particular XSI network. So, send the TSA parameter in xsi network. Multiple TSAs can be configured for single xsi network. TSA configuration has following parameters:

- dampName (optional in tsa): By default, VNFM adds all DAMPs to all TSAs. However, if only specific DAMPs need to be configured, then it can be passed in this parameter.
- fixedIp (optional in tsa): By default, VNFM creates port with dynamic IP based on xsi Network. If fixed IP needs to be configured, then it can be passed in this parameter.
- preferedActiveIPFE (mandatory in tsa): It is based on OptionSets configurations. If IpfeA1IpAddress is configured, then use ipfeA1. Use the following:
  - For IpfeA2IpAddress use ipfeA2
  - For IpfeB1IpAddress use ipfeB1
  - For IpfeB2IpAddress use ipfeB2
- preferedStandByIPFE (mandatory in tsa): Similar to preferedActiveIPFE.

If xsi network is on single subnet, then port for TSA is created only for single IP address. TSA configuration is done only for single IP.

If xsi network is on dual subnet, then port for TSA is created with two IP addresses. Both the IP addresses are configured in TSA.

> **Note:**
>
> - TSA configuration is supported for different combination of DIAMETER flavor.
> - If instantiation of stack has TSA configuration and if scaling operation is done on the same stack, then scaled DAMPs will be added to all the available TSAs.
> - If scaling is performed after discovery operations, then scaled DAMP will not be added to the TSA. This is because discovery operation will not have information about TSA.

Sample Request for Signaling Flavor DIAMETER

Sample request for signaling flavor DIAMETER with TSA configurations with multiple xsi (1, 2, or 4 xsi interface) for dynamic IP (Dual Subnet) deployment model.

Instantiating the first signaling VNF request generated

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


Accept: application/json


Content-Type: application/json

X-Token : <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":[ {
```

```
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
                "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-8.4.0.3.0_85.17.0.vmdk",

                                  "soamAvailabilityZone": "nova",
```

```
                                "ipfeFlavor": "dsr.ipfe",
                                "ipfeImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                "ipfeAvailabilityZone": "nova",
                                "daFlavor": "dsr.da",
                                "daImage": "DSR-8.4.0.3.0_85.17.0.vmdk",
                                "daAvailabilityZone": "nova",
                                "stpFlavor": "dsr.stp",
                                "stpImage":
"DSR-8.4.0.3.0_85.17.0.vmdk",
                                "stpAvailabilityZone": "nova",
                                "soamAffinityPolicy":"anti-affinity",
                                "ipfeAffinityPolicy":"anti-affinity",
                                "daAffinityPolicy":"soft-anti-affinity"
                }
}
```

Instantiating the signaling VNF with TSA configuration with DIAMETER response

```
202 Accepted

Headers:

location: https:// <<VNFM HOST IP>>:8443/vnflcm/v1/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
```

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
  - Tenant 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 obtain 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.

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 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/v1/vnf_instances/<VNF ID received from create request>/instantiate`

```
Accept: application/json

Content-Type: application/json

X-Token: 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": "standy SDS NOAM XMI",
          "resourceId": "5c638770-5585-44c7-97c7-b4a52a26e5ec"
       }
    ],
"vimConnectionInfo":[ {
    "id": "vimid",
    "vimType": "OpenStack",
    "interfaceInfo": {
      "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
    },
    "accessInfo": {
        "username": "dsrci.user",
        "password": "xxxxx",
        "userDomain": "Default",
        "projectDomain": "default",
        "tenant": "DSR CI"
    }

}],
"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",
    "primarySdsNoamVmName": "SDS-NOAM00-ea47f4b1",
```

```
        "sdsDrNoamFlavor": "sds.noam",
        "sdsDrQsFlavor": "sds.noam",
        "sdsDrNoamImage": "SDS-8.4.0.3.0_85.17.0.vmdk",
        "sdsDrQsImage": "SDS-8.4.0.3.0_85.17.0.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": "standy SDS NOAM XMI",
              "resourceId": "5c638770-5585-44c7-97c7-b4a52a26e5ec"
           }
        ],
    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
    "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-8.4.0.3.0_85.17.0.vmdk",
            "sdsDrQsImage": "SDS-8.4.0.3.0_85.17.0.vmdk",
            "sdsDrNoamAvailabilityZone": "nova",
            "sdsDrQsAvailabilityZone": "nova",
            "sdsDrNoamAffinityPolicy":"anti-affinity"
        }
}
```

Sample Response

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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 9-14 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

Table 9-14 (Cont.) Parameters and Definitions SDS DR Network OAM VNF

Parameter	Definitions
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

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 DSR Signaling and XMI IPs from both NOAM VMs.

> **Note:**
>
> The resource IDs can be obtain 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 -> HA** from **Active NOAM GUI**.

- Name of the NOAM SG

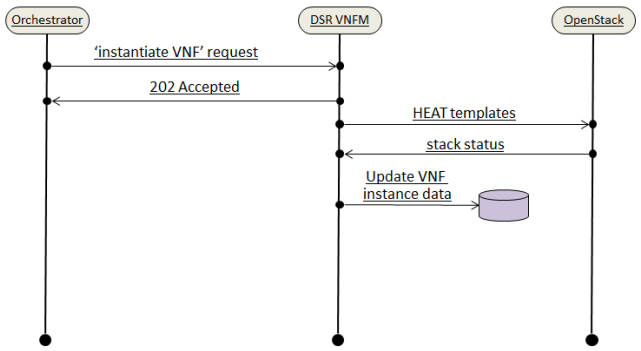
> **Note:**
>
> After SDS deployment, the Max Allowed HA Role of Query Server is expected to be Observer but it is Standby. Manually change the Max Allowed HA Role of Query Server from Standby to Observer as follows:

Login to Active SDS Noam GUI and navigate to **Status & Manage -->HA -->Edit->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 9-5 VNF Create Instance Request



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

Table 9-15 SDS Signaling Flavors supported by VNFM

Signaling Flavors supported by VNFM	Small	Medium	Large
	DP Server	DP Server	DP Server
SDSSIGNALING	1	2	3

The number of DP-SOAM will be 2 for any instantiation level.

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.

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/v1/vnf_instances/ < VNF ID received from create request > /instantiate`

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login
```

Sample request for Dynamic IP deployment model

```
{
                "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":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }
```

```
    }],
                "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-8.4.0.3.0_85.17.0.vmdk",
                                "dpImage": "SDS-8.4.0.3.0_85.17.0.vmdk",
                                "dpSoamAvailabilityZone": "nova",
                                "dpAvailabilityZone": "nova",
                                "dpSoamAffinityPolicy":"anti-affinity",
                                "dpAffinityPolicy":"anti-affinity"
                }
}
```

> **Note:**
>
> The "vipSubnetName" field is used only in case of Dual Subnet.

Sample request for Fixed IP deployment model

```
{
                "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":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
                "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-8.4.0.3.0_85.17.0.vmdk",
                                    "dpImage": "SDS-8.4.0.3.0_85.17.0.vmdk",
                                    "dpSoamAvailabilityZone": "nova",
                                    "dpAvailabilityZone": "nova",
                                    "dpSoamAffinityPolicy":"anti-affinity",
```

```
                                    "dpAffinityPolicy":"anti-affinity"
                }
}
```

Sample Response

```
202 Accepted
Headers:
{
     location: https://localhost:8443/vnflcm/v1/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 9-16 Parameters and Definitions for SDS Sigaling 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 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 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
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

Table 9-16 (Cont.) Parameters and Definitions for SDS Sigaling VNF

Parameters	Definitions
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

Instantiating the ATS Master VNF

The ATS Master VNF Supports dynamic and fixed IP deployment model.

In order to deploy the ATS Master VNF, the following information must be available:

- 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
  - Tenant 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 normally hosts an NTP server, and is often a good choice.

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**.

> **Note:**
>
> It is mandatory to add two XSI Networks in ATS Master to instantiate a stack.

Sample Request for Instantiating ATS Master Dynamic IP 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
X-Token: Token generated after login

{
    "flavourId": "master",
    "instantiationLevelId": "small",
    "extVirtualLinks": "extVirtualLinks",
                "extManagedVirtualLinks": [],

    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }],
    "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_BOX.qcow2",
```

```
        "atsAvailabilityZone": "nova"
    }
}
```

Instantiating ATS Master Request for Fixed IP 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
X-Token: Token generated after login

{
    "flavourId": "master",
    "instantiationLevelId": "small",
    "extVirtualLinks": "extVirtualLinks",
                "extManagedVirtualLinks": [],

    "vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }
    }],
    "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_BOX.qcow2",
        "atsAvailabilityZone": "nova"
    }
}
```

Sample Response

Instantiating the ATS Master VNF response

```
202 Accepted

Headers:
{
     location: https://localhost:8443/vnflcm/v1/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:

Parameter	Definitions
flavourId	Identifier of the VNF deployment flavour to be instantiated
xmiNetwork	Network used to provide access master VM communication
ntpServerIp	IP of the NTP server
dnsServerIp (optional)	Ip of DNS server. If not provided, NTP server IP will be considered as DNS server IP
atsKeyName	key pair name for ATS. To login to 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

Note: The atsKeyName pair is created dynamically through VNFM. Same public key is put into all the ATS instance (master, core & tools) and private key will be in ATS master stack output. Use the same private key to login to ATS instance (master, core & tools), by executing:

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

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

Instantiating the ProvGW VNF

The ProvGW VNF supports dynamic IP deployment model.

In order to instantiate ProvGW, the following information must be available:

- The VNF ID for a previously created ProvGW 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 ProvGW network in the selected OpenStack instance that carries the ProvGW traffic.
- After instantiating VNF ProvGw, a single VM ProvGateway_A is brought up.
- 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**. Swagger specifications can be found post VNFM installation at (`https://<VNFM IP>:8443/docs/vnfm/`).

Sample Request: Sample Request for DYNAMIC IP deployment model

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

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

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

```
"extManagedVirtualLinks": [

],

"vimConnectionInfo":[ {
        "id": "vimid",
        "vimType": "OpenStack",
        "interfaceInfo": {
          "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
        },
        "accessInfo": {
            "username": "dsrci.user",
            "password": "xxxxx",
            "userDomain": "Default",
            "projectDomain": "default",
            "tenant": "DSR CI"
        }

    }]
,
"localizationLanguage": "localizationLanguage",
"additionalParams": {

        "xmi_network" : "ext6-net",

        "ntp": "2606:b400:605:b912:200:5eff:fe00:1f7",

        "image": "UDR-PrvGwy-12.6.0.0.0_18.0.0-dev",

        "no_flavor":"provGw"

}
}
```

Sample Response

```
202 Accepted

Headers:
{
     location: https://localhost:8443/vnflcm/v1/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 9-17 Parameters and Definitions for ProvGW VNF

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
`ntp`	IP of the NTP server
`image` (optional)	Name of image to be used for instantiation of the stack
`no_flavor` (optional)	The Name of the flavor to be used for stack instantiation
`availability_zone` (optional)	The name of the availability zone
`securityGroup` (optional)	The name of the security group

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"
```

```
}
```

Scale VNF to Level (Only Scale Out)

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

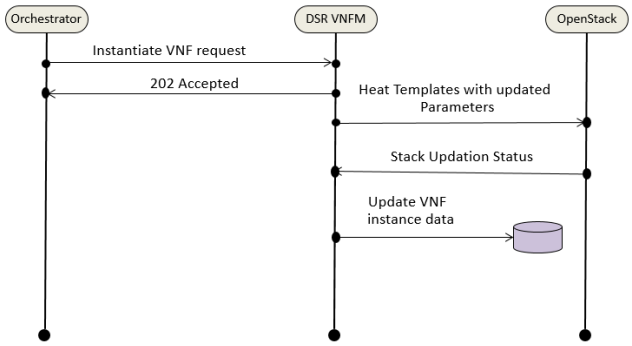
Following are the available options while scaling using "scale to VNF level" N/B Interface:

- Scale VNF to Level based on pre-defined sizes (using Instantiation level Id).
- Scale VNF to Level with arbitrary sizes (using scaleInfo).

> **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.
> - The instantiation level for Signaling stack is available under **Instantiating the first signaling VNF** section.
> - 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 9-6 VNF Scaling



Scale VNF to Level using InstantiationLevelId

This option supports Scaling of VNF from a lower instantiation level to higher one, such as Small to Medium.

Sample Request

Scaling VNF to Level Request for Dynamic IP model

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

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

{

"instantiationLevelId":"medium"

}
```

Scaling VNF to Level Request for Fixed IP model

```
{
  "instantiationLevelId": "medium",
  "additionalParams": {
    "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"],
                                                "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"],
                                                    "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/v1/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

> **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)`
>
> 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>"
               ]
            }
}]
         }
     ]
```

Below table describes the parameters used for sending request to VNFM

Table 9-18 Scaling VNF to Level using InstantiationLevelId

Parameters	Definitions
instantiationLevelId	Identifier of the instantiation level of the deployment flavor to be scaled.
dampXmiIps	List of DAMP external management ips (if new DAMP VMs are to be scaled)
ipfeXmiIps	List of IPFE external management ips (if new IPFE VMs are to be scaled)
stpXmiIps	List of vSTP external management ips (if new vSTP VMs are to be scaled)
sbrXmiIps	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)
dampXsiIps	List of DAMP signaling ips (if new DAMP VMs are to be scaled)
ipfeXsiIps	List of IPFE signaling ips (if new DAMP VMs are to be scaled)
stpXsiIps	List of STP signaling ips (if new DAMP VMs are to be scaled)
dampImiIps	List of DAMP internal management ips (if new DAMP VMs are to be scaled)
ipfeImiIps	List of IPFE internal management ips (if new IPFE VMs are to be scaled)
stpImiIps	List of vSTP internal management ips (if new vSTP VMs are to be scaled)
sbrImiIps	List of SBR internal management ips (if new SBR VMs are to be scaled)
subnet	List of subnet name and ipVersion used (also contains fixed IPs if used)

Note:

During Scaling of SBR's, the newly spawned SBR's are not added to any Server Group, it need to be manually added to the new Server Groups created by the user. One server Group can have maximum two SBR's.

Scale VNF to Level using ScaleInfo (Arbitrary Size)

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/v1/vnf_instances/< VNF ID received from create/instantiate request>/scale_to_level`

```
Accept: application/json
```

```
Content-Type: application/json
```

```
X-Token: Token generated after login
```

Scaling VNF to Level Request for Dynamic IP deployment

```
{
"scaleInfo": [
{
"aspectId": "DAMP",
"scaleLevel": "3"
}],
}
```

Scaling VNF to Level Request for Fixed IP deployment

```
{
  "scaleInfo": [
  {
    "aspectId": "DAMP",
    "scaleLevel": "4"
  },
{
    "aspectId": "IPFE",
    "scaleLevel": "4"
  },
{
    "aspectId": "STPMP",
    "scaleLevel": "4"
  },
{
    "aspectId": "SBR",
    "scaleLevel": "4"
  }
],
"additionalParams": {
"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"],
"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"],
"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/v1/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)`

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 networ>",
 "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>"
                ]
             }
 }]
         }
      ]
```

Below table describes the parameters used for sending request to VNFM.

Table 9-19 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
dampXmiIps	List of DAMP external management ips (if new DAMP VMs are to be scaled)
ipfeXmiIps	List of IPFE external management ips (if new IPFE VMs are to be scaled)
stpXmiIps	List of vSTP external management ips (if new vSTP VMs are to be scaled)
sbrXmiIps	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)
dampXsiIps	List of DAMP signaling ips (if new DAMP VMs are to be scaled)
ipfeXsiIps	List of IPFE signaling ips (if new DAMP VMs are to be scaled)
stpXsiIps	List of STP signaling ips (if new DAMP VMs are to be scaled)
dampImiIps	List of DAMP internal management ips (if new DAMP VMs are to be scaled)
ipfeImiIps	List of IPFE internal management ips (if new IPFE VMs are to be scaled)
stpImiIps	List of vSTP internal management ips (if new vSTP VMs are to be scaled)
sbrImiIps	List of SBR internal management ips (if new SBR VMs are to be scaled)

Note:

During Scaling of SBR's, the newly spawned SBR's are not added to any Server Group, it needs to be manually added to the new Server Groups created by the user. One server Group can have maximum two SBR's.

10

VNF Instantiation across Multi Cloud / Multi Tenant

VNFM supports multi-cloud and multi-tenant deployment for DSR and SDS VNF.

List of VNF deployment of multi cloud/tenant:

Table 10-1 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

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":
"dsrNetworkOam_NOAM_32cd6138_SG",
                                   "vnfInstanceId":
"dsrNetworkOam-4e99a1cd-77b7-478b-9b28-32cd6138"
                }
```

11
Discover Stack

- It is an LCM Discover Rest I/F. This information can be further used by the orchestrator to scale out the stack.
- Before discovering the stack, make sure the following information is available:
  - The Stack ID of the previously created stack.
  - The following information about the OpenStack instance on which the Stack must be discovered:
    - OpenStack Controller URI
    - Use Domain Name
    - Project Domain Id
    - Username
    - Password
    - Tenant name
  - The Interface discovers the stack and performs the following operations:
    - Download the parameter file of the discovered stack.
    - Create the Instance file of the discovered stack.
    - These two files are saved in `/var/vnfm/instances/<autoDiscovery InstanceId>/` directory.

Sample Request for Discover Interface

```
Request URL: POST:
https://<<VNFM HOST IP>>:8443/vnflcm/v1/discover/<<discover stack id>>
For example:
https://localhost:8443/vnflcm/v1/discover/b30ac203-5fe1-4007-
a3ba-078f3422708b
Accept: application/json
Content-Type: application/json
X-Token: Token generated after login
Request Body:
{
  "vimConnectionInfo": [
    {
      "id": "vimid",
      "vimType": "OpenStack",
      "interfaceInfo": {
        "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
      },
      "accessInfo": {
        "username": "dsrat.user",
        "password": "xxxx",
        “userDomain”: “Default”,
```

```
        "projectDomain": "default",
        "tenant": "DSR AT Dev 1"
      }
    }
  ]
}
```

Sample Response for Discover Interface

```
Response Code: 200
{
"vnfInstanceId": "dsrNetworkOam-945cffa107c235bb-43d87678-756b-4f8e-
a59c-d9b7d4dd95a1",
"discoverStackId": "7d861391-0ed2-4d0b-9f01-e84e186e9244"
 }
```

Note:

- Discover VNF Stack supports only the stacks created on VNFM of the same release.
- Discover VNF stack supports only the stack created by VNFM templates of the same release through CLI.
- Discover VNF stack also supports the stack created by VNFM GUI, Double Failure of Active VNFM and its persistent volume.

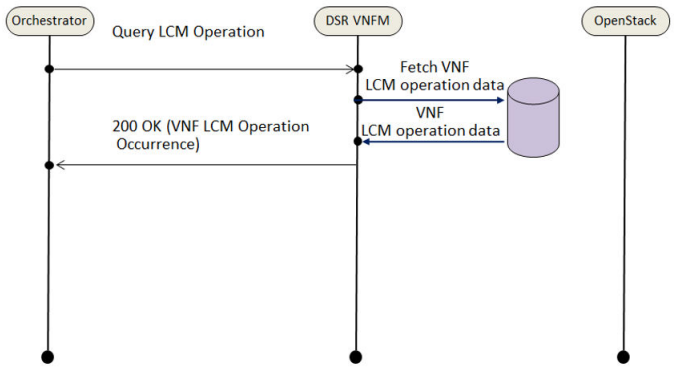
12

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 12-1 VNF LCM Operation



Query LCM Operation, using the following two ways:

- Query individual LCM Operation
- Query All LCM Operation

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 (`vnfLcmOpOccId`) in its resource URI.

The VNFM returns a **200 OK** response to the NFVO, and includes specific data structure of type " `VnfLcmOpOcc`" related to the VNF LCM Operation occurrence identifier (`vnfLcmOpOccId`) in the payload body.

Sample Request

Query individual LCM Operation

URL: `GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_lcm_op_occs / <<{vnfLcmOpOccId}>>`

Sample Response

```
URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_lcm_op_occs/
<<{vnfLcmOpOccId}>>
Accept: application/json
Content-Type: application/json
X-Token: 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": "dsrNetworkOam-cf67bff6-e9c9-4213-b6fa-b5337c3d30b6",
  "operation": "TERMINATE",
  "operationParams": {
    "terminationType": "FORCEFUL",
    "gracefulTerminationTimeout": null,
    "additionalParams": null
  },
  "links": {
    "self": {
      "href": "https://localhost:8443/vnflcm/v1/vnf_lcm_op_occs/
lcmOp-00301ea4-a7b2-4334-8b93-190377700ab0"
    },
    "vnfInstance": {
      "href": "https://localhost:8443/vnflcm/v1/vnf_instances/
dsrNetworkOam-cf67bff6-e9c9-4213-b6fa-b5337c3d30b6"
    }
  },
  "isCancelPending": false,
  "isAutomaticInvocation": false
}
```

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 " `VnfLcmOpOcc`" in the payload body.

Sample Request

Query All LCM Operation

```
URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_lcm_op_occs
```

Sample Response

```
URL: GET: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_lcm_op_occs
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": "dsrNetworkOam-
dfc4dcd2-2752-48b4-875d-6cf703ba4134",
     "operation": "INSTANTIATE",
     "operationParams": {
         "flavourId": "DSR NOAM",
         "instantiationLevelId": "smalll",
         "extVirtualLinks": "extVirtualLinks",
         "extManagedVirtualLinks": [],
         "vimConnectionInfo": [
          {
           "id": "vimid",
            "vimType": "OpenStack",
           "interfaceInfo": {
               "controllerUri": "https://
oortcloud.us.oracle.com:5000/v3"
          },
          "accessInfo": {
             "username": "dsrat.user",
             "password": "xxxxx",
            “userDomain”: “Default”,
               "projectDomain": "default",
             "tenant": "DSR AT Dev 2"
         },
        "extra": null
        }
       ],
       "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/
dsrNetworkOam-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": [
     {
     "id": "vimid",
     "vimType": "OpenStack",
     "interfaceInfo": {
     "controllerUri": "https://oortcloud.us.oracle.com:5000/v3"
    },
    "accessInfo": {
      "username": "dsrat.user",
      "password": "xxxxxx",
      “userDomain”: “Default”,
      "projectDomain": "default",
      "tenant": "DSR AT Dev 2"
    },
    "extra": null
   }
   ],
   "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": "dsrNetworkOam_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
 }
]
```

13
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 13-1 Parameters and Definitions for Terminating VNF

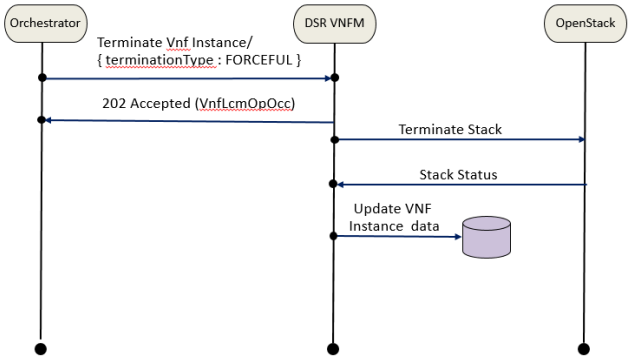
Parameters	Definitions
`terminationType`	Indicates whether forceful or graceful termination is requested.

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 13-1 Forceful Termination



Terminating DSR and SDS VNF Instance Forcefully

Sample Request:

Request URL: `POST: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/<VNF ID received from create request>/terminate`

```
Accept: application/json

Content-Type: application/json

X-Token: Token generated after login

{
   "terminationType": "FORCEFUL"
}
```

Sample Response

```
Response Code: 202
{
     location: https://localhost:8443/vnflcm/v1/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
}
```

Graceful Termination

The VNFM first validates if the VNF configuration is cleaned up after accepting the request. Once that configuration is cleaned, the VNFM deletes the VNF. Then the instance file is updated with VNF Operational State set to **STOPPED**.

If AppWorks configurations are not cleaned manually and the orchestrator tries to do graceful termination for that VNF, then the termination of VNF fails.

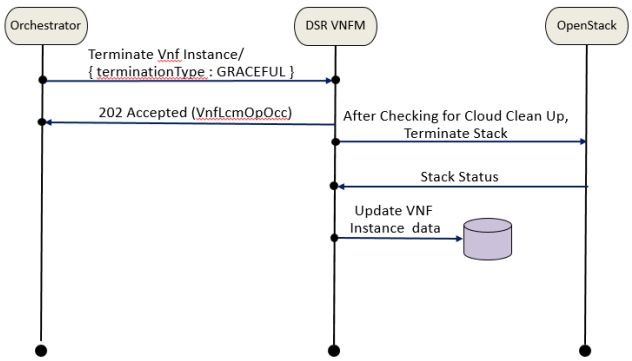
> **Note:**
>
> User must manually cleanup the AppWorks configurations before doing Graceful Termination.

Steps for cleaning up the AppWorks Configuration for Signaling Stack of DSR and SDS:

1. Open corresponding Active NOAM GUI of the Signaling instance.
2. In **Status & Manage** Tab, under **HA**, edit the **Max Allowed HA Role** of instances of the Signaling stack as **OOS**.
3. In Configuration Tab, under Server Groups, edit the corresponding server groups of the instances and uncheck **SG Inclusion** for the Server, and press **OK**. After this step, the excluded Servers must disappear in **Status & Manage -> Server** section.
4. Finally, go to **Configuration -> Servers** section, select the servers that needs to be deleted and click **Delete**.

> **Note:**
>
> For DSR / SDS Signaling VNF clean up, the user must perform the above steps twice, first for C-level servers and then repeat the steps for B-level servers.

Figure 13-2 Graceful Termination



Terminating DSR and SDS VNF Instance Gracefully

Sample Request:

Request URL: `POST: https://<<VNFM HOST IP>>:8443/vnflcm/v1/vnf_instances/<VNF ID received from create request>/terminate`

```
Accept: application/json

Content-Type: application/json
```

```
X-Token: Token generated after login

{
   "terminationType": "GRACEFUL"
}
```

Sample Response

```
Response Code : 202
{
     location: https://localhost:8443/vnflcm/v1/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
}
```

14

Changing the Default Configurations

This section includes information about changing the default configurations through the following files:

- `VmInfo.xml`
- `VnfmProperties.xml`

Changing Flavor Names

To change the flavor names:

1. Log into the VNFM VM.
2. Go to `/opt/vnfm/config/8.4/` folder.
3. Edit the file `VmInfo.xml`
4. Find the tag `<flavor>` against the VM type (NOAM, SOAM, and so on)
5. Change the default name to user defined name.

> **Note:**
>
> The user defined flavor name should be a valid flavor.

Changing Image Names

1. Log into the VNFM VM.
2. Change to `/opt/vnfm/config/8.4/` folder.
3. Edit `VmInfo.xml`.
4. Find the tag `<image>` against the VM type (NOAM, SOAM, and so on).
5. Change the default name to user defined name.

> **Note:**
>
> The user defined image name should be a valid image.

Changing Availability Zone

1. Log into the VNFM VM.
2. Change to `/opt/vnfm/config/` folder

3. Edit the `VnfmProperties.xml`
4. Find the tag `<osAvailabilityZone>`
5. Change the default name to user defined name.

> **Note:**
>
> The user defined flavor name should be the availability zone.

Changing Profile Name

1. Log into the VNFM VM.
2. Change to `/opt/vnfm/config/` folder.
3. Edit `VnfmProperties.xml`.
4. Find the tag `<profileName>`.
5. Change the default name to user defined name.

> **Note:**
>
> The user defined image name should be a valid profile name.

15
VNFM SNMP ALERTS

- VNFM supports both Single and Dual SNMP Manager for SNMP alerts.
- VNFM acts as an SNMP Agent that generates traps such as exception traps, and success notifications.
- VNFM MIB (`oracleVnfm.mib`) and Top level MIB (`tekelec-toplevel-reg.mib`) are placed in "`/usr/share/snmp/mibs`" directory.
  For more information on Alerts and MIB files, see sections VNFM Alarms and VNFM MIB File.
- VNFM generates traps in the following SNMP versions:
  - System health traps - SNMP v2c version
  - VNFM exception and success notifications - SNMP v3 version

VNFM IP (eth0)	VNFM IP (eth1)	SNMP Manager 1	SNMP Manager 2	Supported
IPV4	IPV4	IPV4	IPV4	Yes
		IPV6	IPV6	N/A
		IPV6	IPV4	N/A
		IPV4	IPV6	N/A
IPV6	IPV4	IPV4	IPV4	Yes
		IPV4	IPV6	Yes
		IPV6	IPV4	Yes
		IPV6	IPV6	Yes

See section 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/Manager

To change the SNMP Trap Receiver/Manager:

1. Edit the IP and port of SNMP Trap Receiver/Manager by changing the property `<address>ip/port</address>` inside `SnmpReceiverIPs` node of `VnfmProperties.xml` file, located in `/opt/vnfm/config/` 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**:

   ```
   <SnmpReceiverIPs>
     <address>2606:b400:605:b813::5/7400</address>
   </SnmpReceiverIPs>
   ```

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/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 we need to add rule in Security group like below format of VNFM stack. EX:

1. Add second address inside `SnmpReceiverIPs` property as `<address>ip/port</address>` in `/opt/vnfm/config/VnfmProperties.xml`.
   For example:

   ```
   <SnmpReceiverIPs>
       <address>10.75.189.151/8900</address>
       <address>2606:b400:605:b813::5/7400</address>
   </SnmpReceiverIPs>
   ```

2. Add rule in Security group in VNFM stack format, for second SNMP Manager receiver port other than 162.
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/Manager.

SNMP System Traps Configurations

SNMP System traps have some default configurations specified in the file:

```
/var/vnfm/prometheus/alertmanager/alertmanager.yml


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, please following the following steps:

In `dsrvnfm` user mode:

- Edit the `group_interval` or `repeat_interval` time configuration in this file:

  `/var/vnfm/prometheus/alertmanager/alertmanager.yml`

- Execute the script:

  `/var/vnfm/prometheus/snmp_notifier/restart_SnmpNotifier.sh <VNFM IP Address>`

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 15-1 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
3012	GEN_012	vnfmInterruptedErrorAlertNotification	An interrupted error has occurred	Minor

Table 15-1 (Cont.) General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
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 15-1 (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	vnfmUnsopportedFlavorIdAlertNotification	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

Table 15-1 (Cont.) General Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
3064	GEN_064	vnfmUnsupportedEncodingAlertNotification	Unsupported Encoding Found	Minor
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 occured while updating the Remote VNFM State Info.	Minor

Example OID: `.1.3.6.1.4.1.323.5.3.33.1.2.2.4001`

Table 15-2 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

Table 15-2 (Cont.) Semantic Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
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
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	vnfmMultipleOccurenceOfParameterAlertNotification	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

Example OID: `.1.3.6.1.4.1.323.5.3.33.1.2.3.5001`

Table 15-3 OpenStack Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
5001	OPENSTACK_001	vnfmClientCreateFailureAlertNotification	Failed to create Openstack Client.	Minor

Table 15-3 (Cont.) OpenStack Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
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_007	vnfmStackNotFoundErrorAlertNotification	Failed to find the stack by the name	Minor
5007	OPENSTACK_008	vnfmStackCreateServerErrorAlertNotification	Failed to perform stack create operation due to error on server	Major
5008	OPENSTACK_009	vnfmStackGetOutputsTimeoutErrorAlertNotification	Failed to retrieve a stack infrastructure	Minor
5009	OPENSTACK_010	vnfmStackGetOutputsConfigErrorAlertNotification	Failed to open NsaOsProperties file	Minor
5010	OPENSTACK_011	vnfmStackGetOutputsMissingDataErrorAlertNotification	Required data missing from getOutputs response	Minor
5011	OPENSTACK_012	vnfmStackGetOutputsNullValueErrorAlertNotification	Failed to retrieve IPs from stack	Minor
5012	OPENSTACK_013	vnfmInvalidJsonFormatErrorAlertNotification	The generated JSON String has errors	Minor
5013	OPENSTACK_014	vnfmOpenstackCliCommandExecutionFailureAlertNotification	OpenStack command execution failure	Minor
5014	OPENSTACK_015	vnfmStackServiceConfigErrorAlertNotification	Error just before stack creation	Minor
5015	OPENSTACK_016	vnfmStackUpdateClientErrorAlertNotification	Failed to perform stack update operation due to error on client	Major
5016	OPENSTACK_017	vnfmStackUpdateServerErrorAlertNotification	Failed to perform stack update operation due to error on server	Major
5017	OPENSTACK_018	vnfmStackDeleteServerErrorAlertNotification	Failed to perform stack delete operation due to error on server	Major
5018	OPENSTACK_019	vnfmNetworkDetailsNotFoundAlertNotification	Failed to fetch the network details from the provided network	Minor
5019	OPENSTACK_020	vnfmIpDetailsNotFoundAlertNotification	Failed while fetching IP details for the provided resource ID	Minor
5020	OPENSTACK_021	vnfmPortCreationErrorAlertNotification	Failed while creating port from network ID.	Minor
5021	OPENSTACK_022	vnfmNetworkNameFromIdAlertNotification	Failed while fetching network name from network ID.	Minor
5022	OPENSTACK_023	vnfmStackDetailsNotFoundAlertNotification	Failed while fetching stack output from stack.	Minor

Example OID:.1.3.6.1.4.1.323.5.3.33.1.2.4.6001

Table 15-4 Invalid Gen Exception Alert Summary

OID	Alert ID	Alert Name	Alert Message	Severity
6001	Invalid_GEN_001	vnfmIncorectVnfInstanceIdAlertNotification	Incorrect Vnf Instance ID	Minor
6002	Invalid_GEN_002	vnfmIncorrectrStackIdOrNameAlertNotification	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	vnfmIncorrectVnfLcmOpOddIdAlertNotification	Incorrect VNF LCM Operation Occurrence Id	Minor

Example OID:`.1.3.6.1.4.1.323.5.3.33.1.2.5.7001`

Table 15-5 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
7003	STATE_CONFLICT_003	timeStampTooOldAlertNotification	Request too old	Minor

Example OID:`.1.3.6.1.4.1.323.5.3.33.1.2.6.8001`

Table 15-6 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	vnfmStackCreateSucessAlertNotification	Info
8002	02	STACK UPDATE	The vnfm Operation Stack Update is successful	vnfmStackUpdateSucessAlertNotification	Info
8003	03	STACK DELETE	The vnfm Operation Stack Terminate is successful	vnfmStackDeleteSucessAlertNotification	Info
8004	04	STACK Discovery	The vnfm Operation Stack Discover is successful	vnfmStackDiscoverSucessAlertNotification	Info
8005	05	CLOUD INIT	The vnfm Operation Cloud-Init is successful	vnfmCloudInitSucessAlertNotification	Info
8006	06	Remote Synchronization	Remote Synchronization is successful.	vnfmRemoteSynchronizationSuccessAlertNotification	Info

Table 15-6 (Cont.) VNFM Success Alert

OID	Success Alert ID	Operation	Success Alert Message	Alert Name	Severity
8007	07	Change VNFM State	Change VNFM State is successful	changeVnfmStateInfoSuccessAlertNotification	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	vnfmDiskUageAlertNotification	Warning

Example OID:.1.3.6.1.4.1.323.5.3.33.1.2.8.9001

Table 15-7 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 MIB File

Oracle VNFM MIB file for exceptions:

```
--
-- VNFM 4.3.0
-- Copyright (C) 2019, 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 & Canada: 888.367.8552
         India: +91.124.436.8552
         China: +65.6248.4510
         UK & 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 {
                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."
  ::= { 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."
  ::= { vnfmSucessAlertEntry 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 }


vnfmSucessAlertNotificationsGroup 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 }

--
-- 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."
  ::= { vnfmInstanceDownAlertNotification 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."
  ::= { vnfmInstanceDownAlertNotification 3 }

vnfmMemoryUsageAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmMemoryUsageAlertName, vnfmMemoryUsageAlertSeverity,
vnfmMemoryUsageAlertDescription }
```

```
     STATUS       current
     DESCRIPTION
         "Alert for High Memory Usage."
     ::= { vnfmSystemAlertNotificationsGroup 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."
  ::= { vnfmMemoryUsageAlertNotification 1 }

vnfmMemoryUsageAlertSeverity OBJECT-TYPE
  SYNTAX    OCTET STRING
  MAX-ACCESS read-only
  STATUS    current
  DESCRIPTION
    "The severity of the alert being sent."
  ::= { vnfmMemoryUsageAlertNotification 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."
  ::= { vnfmMemoryUsageAlertNotification 3 }

vnfmLoadAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmLoadAlertName, vnfmLoadAlertSeverity,
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."
  ::= { vnfmLoadAlertNotification 1 }

vnfmLoadAlertSeverity OBJECT-TYPE
  SYNTAX    OCTET STRING
  MAX-ACCESS read-only
  STATUS    current
```

```
  DESCRIPTION
    "The severity of the alert being sent."
  ::= { vnfmLoadAlertNotification 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."
  ::= { vnfmLoadAlertNotification 3 }

vnfmDiskUageAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmDiskUageAlertName, vnfmDiskUageAlertSeverity,
vnfmDiskUageAlertDescription }
     STATUS       current
     DESCRIPTION
         "Alert for high Disk Usage."
     ::= { vnfmSystemAlertNotificationsGroup 2004 }

vnfmDiskUageAlertName 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."
  ::= { vnfmDiskUageAlertNotification 1 }

vnfmDiskUageAlertSeverity OBJECT-TYPE
  SYNTAX    OCTET STRING
  MAX-ACCESS read-only
  STATUS    current
  DESCRIPTION
    "The severity of the alert being sent."
  ::= { vnfmDiskUageAlertNotification 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 }

vnfmHttpResourceNotFoundAlertNotification 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 }

vnfmUnsopportedFlavorIdAlertNotification 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 }

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 occurence 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 }

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 }

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 }


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 }

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."
     ::= { vnfmSucessAlertNotificationsGroup 8004 }

vnfmCloudInitSucessAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmSuccessAlertID, vnfmOperation,
vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
     STATUS       current
     DESCRIPTION
         "The Cloud Init is successful."
     ::= { vnfmSucessAlertNotificationsGroup 8005 }

vnfmRemoteSynchronizationSuccessAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmSuccessAlertID, vnfmOperation,
vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
     STATUS       current
     DESCRIPTION
         "Remote Synchronization is successful."
     ::= { vnfmSucessAlertNotificationsGroup 8006 }

changeVnfmStateInfoSuccessAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmSuccessAlertID, vnfmOperation,
vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
     STATUS       current
     DESCRIPTION
         "Change VNFM State is successful."
     ::= { vnfmSucessAlertNotificationsGroup 8007 }

vnfmInvalidUserScopeAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmSuccessAlertID, vnfmOperation,
vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
     STATUS       current
     DESCRIPTION
         "Scope not allowed for this user."
     ::= { vnfmAuthExceptionAlertNotificationsGroup 9001 }

vnfmUserAlreadyPresentAlertNotification NOTIFICATION-TYPE
     OBJECTS { vnfmSuccessAlertID, vnfmOperation,
vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
```

```
        STATUS       current
        DESCRIPTION
            "User Already Present."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9002 }

    vnfmInvalidCredentialsEnteredAlertNotification NOTIFICATION-TYPE
        OBJECTS { vnfmSuccessAlertID, vnfmOperation,
    vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
        STATUS       current
        DESCRIPTION
            "Invalid username or password entered."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9003 }

    vnfmSessionExpiredAlertNotification NOTIFICATION-TYPE
        OBJECTS { vnfmSuccessAlertID, vnfmOperation,
    vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
        STATUS       current
        DESCRIPTION
            "Session Expired, please login again to continue."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9004 }

    vnfmInvalidTokenAlertNotification NOTIFICATION-TYPE
        OBJECTS { vnfmSuccessAlertID, vnfmOperation,
    vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
        STATUS       current
        DESCRIPTION
            "Invalid Token."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9005 }

    vnfmNullTokenAlertNotification NOTIFICATION-TYPE
        OBJECTS { vnfmSuccessAlertID, vnfmOperation,
    vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
        STATUS       current
        DESCRIPTION
            "Token Field must be present."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9006 }

    vnfmInvalidStateAlertNotification NOTIFICATION-TYPE
        OBJECTS { vnfmSuccessAlertID, vnfmOperation,
    vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
        STATUS       current
        DESCRIPTION
             "VNFM State is invalid."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9007 }

    vnfmUnauthorizedAccessAlertNotification NOTIFICATION-TYPE
        OBJECTS { vnfmSuccessAlertID, vnfmOperation,
    vnfmSucessAlertMessage, vnfmSucessAlertTimeStamp, vnfmSuccessSeverity }
        STATUS       current
        DESCRIPTION
            "Unauthorized Access."
        ::= { vnfmAuthExceptionAlertNotificationsGroup 9008 }
    END
```

16

Import HTTPS/SSL Certificate into VNFM

> **Note:**
>
> Diameter must be configured for running the traffic.

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.

Execute 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.

> **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.

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.

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 executing:
`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 executing:

   `openssl req -noout -text -in sslcert.csr | grep DNS`

3. Generating a self-signed certificate:
   To generate a temporary certificate, which is acceptable for 365 days, execute:

   `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 executing:

   `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 executing:
```
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]
```

To delete existing alias from the keystore file, execute (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.

17
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. To check if the certificate has been added into the keystore file successfully, execute:

   ```
   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. To delete certificates manually from the keystore, execute:

   ```
   keytool -delete -keystore <NAME OF THE KEYSTORE FILE> -store pass
   <PASSWORD OF THE KEYSTORE> -alias <keyAlias>
   ```

> **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.

Configurable Keystore

There is a script named `configKeystore.py`, which is present in the project directory at `/vnfm/src/main/resources/com/oracle/dsr/vnfm/install/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.

18
NOAM IPv6 Migration

Prerequisite: The xmi & 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 18-1 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, incase a subnet migrate is required, then perform the following steps manually.

Steps to migrate DSR NOAM on IPv6:

1. Create DSR Noam through VNFM. Provide dual subnet network (xmi & imi) to creating the DSR Noam set up. DSR NOAM will be up & 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.

   Execute 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 document *Dual IP Stack migration* to add the IPv6 interface for active/standby NOAM xmi, imi and VIP.
4. While adding IPv6 interface, use the same IPv6 IP address for active/standby xmi & 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 & vip port.
   c. Open the port to obtain the created IPv4 & IPv6 address.

19

Troubleshooting VNFM

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 "/usr/share/vnfm/apache-tomcat-9.0.16/logs/catalina.out".
- Tomcat logs are located in " /usr/share/vnfm/apache-tomcat-9.0.16/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"

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"

Adding Route for a New VIM

To add route for a new VIM, execute the following commands in `root` user mode:

1. Open `route-network.sh`, and append the new VIM route address to the DataList. For example: `DataList=10.75.167.0/24,10.75.185.0/24`
2. Execute `ifdown eth1`, and then `ifup eth1`

Reboot Tomcat

To reboot Tomcat, execute the following commands in `'dsrvnfm'` user mode:

1. `/usr/share/vnfm/apache-tomcat-9.0.20/bin/shutdown.sh`
2. `/usr/share/vnfm/apache-tomcat-9.0.20/bin/startup.sh`

Resolve HA Alarms on VNF through VNFM Deployed Setup

Perform the following 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 executing:
   `tcpdump -i eth1 -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 19-1 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 executing:
   `tcpdump -i eth1 -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 executing:
   `pm.set off cmha && sleep 5 && pm.set on cmha`

> **Note:**
>
> If the Node is HA Active, then restarting cmha will cause switch over.

Adding a Port in Openstack Security Groups

The Security Group Rules define the traffic that is allowed through instances assigned to the security group.

To allow traffic through ports other than the default ports added by VNFM, execute:

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**.

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.

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
```