VNF Installation and Upgrade Guide

Contents

Chapter 1. About this document

The *Smart Plan Suite (SPS) Installation Guide* describes the installation process and procedures for image-based, distributed deployment of the SPS.

Intended audience

The intended audience of this document includes:

- **Service provisioners** who provision the SPS data.
- **Service administrators** who monitor the SPS.
- **Field support personnel** who install the SPS and upgrade it to later versions.

Used conventions

The following table describes the conventions.

Table 1. Conventions

Appearance	Description
emphasis	Text that is emphasized.
graphical user interface text	Text that is displayed in a graphical user interface:
`system input`	Text that the user types as input to a system
`system output`	A value or command-line parameter that the user provides.
variable	A value or command-line parameter that the user provides
[ ]	Text or a value that is optional.
{ `value1` | `value2` } { *variable1* | 8.7 *variable2* }	A choice of values or variables from which one value or variable is used

General information

The following table shows the documentation resources that may be of use when working with the SPS.

Table 2. SPS 20.6 Documentation

Title	Part Number
Overview Guide	9YZ-09126-UG00-PCZZA
Installation Guide	9YZ-09126-IN00-RJZZA

Title	Part Number
Operation, Administration, and Maintenance (OAM) Guide	9YZ-09126-MT00-REZZA
Policy User Guide	9YZ-09126-UG03-PCZZA
Charging User Guide	9YZ-09126-UG01-PCZZA
Subscriber Manager Guide	9YZ-09126-UG05-PCZZA
Notification Service (NS) User Guide	9YZ-09126-UG06-PCZZA
Global Configuration Guide	9YZ-09126-CN02-TCZZA
Diameter Configuration Guide	9YZ-09126-CN01-TCZZA
Use Cases Guide	9YZ-09126-PL00-FPZZA
Service Manager API Reference	9YZ-09126-UG04-EFZZA
HLAPI Reference	9YZ-09126-CN06-PEZZA
Standards Compliance Reference	9YZ-09126-CN00-EFZZA
Error Messages Reference	9YZ-09126-AL00-QEZZA
CDR/EDR Reference	9YZ-09126-MT01-REZZA
Free and Open Source Software (FOSS) Guide	9YZ-09126-PL02-QEZZA
Privacy Consideration Guide	9YZ-09126-PL03-QEZZA
Data Model Specification for CDM (SPS)	9YZ-ONENDS-CN03-PEZZA
One-NDS SPML Provisioning ISPEC	9YZ-ONENDS-CN04-PEZZA
Provisioning Gateway Application Extension Package (ONENDS-WX-SPS) Release Note	9YZ-ONENDS-PL04-FPZZA
VNF Artifacts Generator Tool User Guide	9YZ-09126-CN05-TCZZA
DevOps Guide	9YZ-09126-MT02-REZZA
Engineering Guidelines	9YZ-09126-CN08-PEZZA
Glossary	9YZ-09126-UG06-WDZZA

Document support

Customer documentation welcome page

- https://documentation.nokia.com/

Technical support

- https://customer.nokia.com/support/s

How to comment

Documentation Feedback

- documentation.feedback@nokia.com

Chapter 2. SPS installation prerequisites

This chapter describes the prerequisites for the installation of the Smart Plan Suite (SPS).

SPS and cloud environment

Overview

Note:

You should read the Overview Guide before installing the SPS. The Overview Guide provides the definition of the data model and its attributes within the SPS.

The SPS is a cloud-native, scalable, high-performance, and feature-rich product that supports flexible network configurations with multi-site deployments and an integrated suite of software components.

The SPS is deployed with three types of sites: Service Manager (SM), Managed Element (ME) and SM+ME in the same VNF (This site type is called IG). The SM site is used for provisioning and API access. The ME site is used for SPS Charging, Policy and Notification Server (NS) applications. The IG site combines all the features of SM and ME within a single site. For more information on the site types and their components, see the Overview Guide.

The SPS supports OpenStack and VMware but has no specific dependency on either of them. The SPS is agnostic to cloud platforms and thus it is capable of adapting to new platforms. Some SPS components such as NS support containers and microservices. The SPS uses the Network Function Virtualization (NFV) cloud model that hosts, deploys and services Virtual Network Functions (VNFs).

SPS integration with Nokia CloudBand software

The SPS is integrated with Nokia CloudBand software products: CloudBand Infrastructure Software (CBIS) and CloudBand Application Manager (CBAM). CBAM and CBIS follow the ETSI NFV model and NFV Management and Orchestration (NFV-MANO) architectural framework. The SPS is managed by CBAM and can be deployed on the following cloud platforms: CBIS with OpenStack, OpenStack and VMware vCloud.

CBIS (OpenStack)

CBIS is an ETSI NFV MANO system which is a Virtual Infrastructure Manager (VIM) and a Network Function Virtualization Infrastructure (NFVI) suite for OpenStack. CBIS can be flexibly deployed in any combination of NFVI and VIM roles. CBIS virtualizes and manages compute, storage, and networking resources to enable VNFs to run. CBIS performs the OpenStack-based VIM function and provides virtualization software (hypervisor, virtual switch, monitoring) installed on each server. CBIS virtualizes storage resources across servers and supports external storage arrays as well as enhanced platform awareness (EPA) for high-performance VNFs. In addition, it provides

tools for monitoring configuration and troubleshooting. CBIS is pre-integrated and validated with a set of hardware components, including Nokia AirFrame.

CBAM

CBAM is an ETSI NFV Generic Virtualized Network Function Manager (G-VNFM) designed to run in OpenStack, vCloud or NCIV environment. CBAM uses Heat Orchestration Templates (HOT) to manage OpenStack VNFs, and Open Virtualization Format (OVF) to manage VMware VNFs (with vCloud API). The key functionality of CBAM is the VNF lifecycle management (LCM). In CBAM the lifecycle workflow orchestration is performed by the VNF Topology and Lifecycle Manager (TLM) with the actual workflow execution steps carried out in Mistral (a workflow execution engine responsible for the workflow steps, and visibility and control over the execution process). Ansible is used to perform various configuration actions related to the LCM operations.

The following figure shows the basic VNF LCM operations that use built-in workflows. The SPS provides custom and automated LCM operations for upgrade, heal and scale.

Figure 1. VNF lifecycle

Before installing SPS

Overview

Note:

Each SM, ME and IG site is associated with one VNF and one stack, including nested stacks.

Before installing the SPS, do the following:

- Determine the number of SM, ME and IG sites needed for the SPS deployment, whether or not Geo-redundancy is applicable to any of the sites in the deployment as well as the number of VMs needed for each site.
- Configure network. For more information, see SPS and networks *(on page 6)*.
- Install and configure CBAM. For more information, see CBAM configuration *(on page 12)*.
- Install and configure cloud platforms: VMware vCloud NFV or OpenStack.

- If CBIS is used for OpenStack deployment, install and configure CBIS. For more information, see CBIS installation and configuration *(on page 6)*.
- Configure cloud-platform resources and services. For more information, see SPS cloud-platform resources and services *(on page 8)*.

CBIS installation and configuration

To use CBIS as infrastructure for the SPS, network, firewall and quotas must be configured as required by CBIS. For information on CBIS, including architecture, required hardware and installation, see General information *(on page 1)*.

SPS and networks

The SPS supports internal and external networks.

Note:

The default internal and external networks in the VNF_CONF files cannot be deleted. These default networks can be modified but only as described in the SPS VNF Artifacts Generator Tool Guide.

The success of the SPS installation is critically dependent on the network. The network must be planned and configured before the SPS deployment. When designing the network infrastructure, do the following:

- Define the number of SM sites, ME sites and IG sites.
- Consider the networks for multiple services such as
  - Operation Administration and Management (OAM)
  - Geo-redundancy for DB VMs
  - Diameter traffic on IOHD (Diameter Load Balancer) VMs
  - LDAP provisioning traffic on IOHO (HTTP and TCP Load Balancer) VMs

Note:

- For OpenStack deployments, Neutron is used as networking service.
- Although NAT is used internally and transparent to users, NAT is not used externally.

SPS internal network

The SPS internal network is a single network that uses IPv4 only. If there is a conflict between the default IP addresses and your network, replace the internal network host identifier by the new x.y.z value. If you change the default IP addresses, you must set the new values as follows:

- <internal network host identifier>.20
- <internal network host identifier>.21
- <internal network host identifier>.22
- <internal network host identifier>.23
- <internal network host identifier>.24

The internal network host identifier can be set to any x.y.z value. However, it is highly recommended to use a private IP range (as defined by RFC1918) since the internal network host identifier is used as a backplane internal network. To change the internal network host identifier, choose any x.y.z values within the following ranges but ensure that they do not conflict with any other external subnet used by your network:

- 10.0.0-10.255.255
- 172.16.0-172.31.255
- 192.168.0-192.168.255

There is no restriction on the use of the same internal network identifier on different sites in the SPS. The same internal network identifier can be used by all the sites in the SPS.

Note:

If the internal network host identifier is changed, ensure that this change is consistent with all the applicable IP addresses in the VNF Extension file.

The /24 internal subnet should be reserved networkwide for the designated use by the SPS only.

SPS external networks

The SPS external network can be a single network or it can consist of multiple networks, and use both IPv4 and IPv6. The SPS supports assigning IP addresses to external interfaces of each VM. However, the SPS supports only static IP address assignment. All the external virtual IP addresses and fixed IP addresses must be statically assigned.

When defining IPv6 subnets, ensure that the number of these subnets is equal or greater than the number of ComSvc VMs to be deployed. A full /80 IPv6 subnet must be reserved for each ComSvc VM. Each /80 IPv6 subnet must be a slice of the external IPv6 subnet attached to the ComSvc VMs. Each ComSvc VM uses its reserved /80 IPv6 subnet to assign IPv6 addresses to local containers.

Note:

Since IETF strongly recommends to use fixed /64 bits prefix-length when assigning IPv6 addresses, the IPv6 prefix-length assigned to any SPS node's external IPv6 interfaces must be /64. The other IPv6 length prefixes are not supported. For more information, see RFC 7421.

Note:

When planning the SPS network, ensure that the communication between CBAM and OAME will be available and the CBAM VM will be able to communicate with both OAME VMs (SSH over IPv4 or IPv6 depending on the network configuration used).

Supported network combinations

The following table describes the network combinations supported by the SPS.

Table 3. Supported network combinations

Network combination	Internal network	External networks			
	IPv4	IPv4		IPv6	
	Single network	Single network	Multiple networks	Single network	Multiple networks
combination #1	yes	yes	no	no	no
combination #2	yes	no	yes See the note following this table.	no	no
combination #3	yes	no	no	yes	no
combination #4	yes	no	no	no	yes See the note following this table.
combination #5	yes	yes	no	yes	no
combination #6	yes	no	no	no	no
combination #7	yes	no	yes	no	yes

Note:

If a VM has two or more public IP interfaces of the same IP version, the static IP routes for that IP version need to be identified and updated when the CSAR package is generated. The static IP routes are automatically configured during the VNF instantiation. To configure static routes on the site that has been instantiated, contact Nokia Professional Services.

The following table lists the network combinations available for all the VM types. However, for each VM type only one combination should be chosen. For example, out of six combinations available for IOHO VMs, only one must be chosen.

Table 4. Network combinations available for VMs

VM type	Network combinations
IOHO	combinations: #1, #2, #3, #4, #5, #7
OAME	combination #1, #2, #3, #4, #5, #7
DB	combination #1, #3
Auxiliary	combination #6
SM App	combination #3, #6
IOHD	combination #1, #2, #3, #4, #5, #7
ComSvc	combination #1, #3, #5
Diameter App	combination #3, #6
CDR	combination #1, #3, #5

SPS cloud-platform resources and services

The cloud-platform resources that are used by the SPS such as Nova flavors and Cinder volumes for OpenStack, and Independent disk for VMware have to be configured according to SPS Dimensioning and Engineering Guidelines. For assistance, contact Nokia Professional Services.

The following table describes the minimum SPS configuration. For more information on SPS configurations, see the Overview Guide.

Table 5. Minimum SPS configuration

VNF	VM name segment	Number of VMs based on scalability factor or cluster size (for Auxiliary)	vCPUs	Memory (RAM) in GB	Root disk in GB	Minimum storage volume per VM in GB
SM	ioho	$2_{(1+1)}$	2	6	60	11
	oame	$2_{(1+1)}$	4	6	60	30
	db	$4_{2*(N+K)}$	2	6	60	22
	auxiliary	5	4	8	80	20
	smapp	$2_{(N+K)}$	2	6	60	21
ME	ioho	$2_{(1+1)}$	2	6	60	11
	iohd	$2_{(1+1)}$	2	6	60	11
	oame	$2_{(1+1)}$	4	6	60	30
	comsvc	$2_{(N+K)}$	4	24	80	21
	db	$4_{2*(N+K)}$	2	6	60	22
	diameterapp	$2_{(N+K)}$	2	6	60	11
	cdr	$2_{(N+K)}$	2	6	60	21
	auxiliary	5	4	8	80	20
IG	ioho	$2_{(1+1)}$	2	6	60	11
	iohd	$2_{(1+1)}$	2	6	60	11
	oame	$2_{(1+1)}$	4	6	60	30
	comsvc	$2_{(N+K)}$	4	24	80	21
	db	$4_{2*(N+K)}$	2	6	60	22
	diameterapp	$2_{(N+K)}$	2	6	60	11
	cdr	$2_{(N+K)}$	2	6	60	21
	auxiliary	5	4	8	80	20
	smapp	$2_{(N+K)}$	2	6	60	11

Storage configuration

The following table lists the volumes used for storage for different VM types.

Table 6. Volume types used for various VM types

VM type	Volume type
Auxiliary	Data and log
OAME	Data, backup and log
DB	Data, datastore, log and digest log
IOHO	Data and log
IOHD	Data and log
ComSvc	Data, Kafka and log
Diameter App	Data and log
CDR	Data, log and record
SM App	Data, Kafka and log

The following table describes the volumes used for storage. For OpenStack deployments, these volumes apply to Cinder volumes. For VMware deployments, these volumes apply to Independent disk.

Note:

- Storage volumes are configured before instantiation and as described in the SPS VNF Artifacts Generator Tool Guide.
- Storage volumes must be configured according to SPS Dimensioning and Engineering Guidelines.

Table 7. Storage volumes

VM name segment	Volume name	Description	Applies to	Minimum storage volume per VM in GB
auxiliary	auxiliary_data_volume	This volume is used on Auxiliary VMs. This volume is mounted on /appdata directory. Zookeeper data is stored at this mountpoint.	ME SM IG	10
	auxiliary_log_volume	This volume is used on Auxiliary VMs. This volume is mounted on /var/log directory.	ME SM IG	10
db	db_data_volume	This volume is used on DB VMs. This volume is mounted on /appdata directory.	ME SM IG	1
	db_datastore_volume	This volume is used on DB VMs. The mountpoint is /opt/tpa/datastore/dsc where Aerospike data is stored.	ME SM IG	10
	db_log_volume	This volume is used on DB VMs. This volume is mounted on /var/log directory.	ME SM IG	10
	db_digestlog_volume	This volume is used to on DB VMs. The mountpoint is defined by <aerospike/xdr/xdrdigestlogpath/location> parameter in the SPS section of the VNF Extension file. The Aerospike XDR digestlog is stored at this mountpoint.	ME SM IG	1

VM name segment	Volume name	Description	Applies to	Minimum storage volume per VM in GB
ioho	ioho_data_volume	This volume is used on IOHO VMs. This volume is mounted on /appdata directory.	ME SM IG	1
	ioho_log_volume	This volume is used on IOHO VMs. This volume is mounted on /var/log directory.	ME SM IG	10
oame	oame_data_volume	This volume is used on OAME VMs for Prometheus data, and system and application metrics. This volume is mounted on /appdata directory.	ME SM IG	10
	oame_backup_volume	This volume is used on OAME VMs for backup and recovery. Several mountpoints (for example, /BACKUP) are used for the storage of backup, and recovery data and files.	ME SM IG	10
	oame_log_volume	This volume is used on OAME VMs. This volume is mounted on /var/log directory.	ME SM IG	10
cdr	cdr_data_volume	This volume is used on CDR VMs. This volume is mounted on /appdata directory.	ME IG	1
	cdr_log_volume	This volume is used on CDR VMs. This volume is mounted on /var/log directory.	ME IG	10
	cdr_record_volume	This volume is used for charging data records on CDR VMs. This volume is mounted on /cdr directory.	ME IG	10
comsvc	comsvc_data_volume	This volume is used on ComSvc VMs. This volume is mounted on /appdata directory.	ME IG	1
	comsvc_kafka_volume	This volume is used on ComSvc VMs. The mountpoint is /kafka_data where Kafka data is stored.	ME IG	10
	comsvc_log_volume	This volume is used on ComSvc VMs.	ME	10

VM name segment	Volume name	Description	Applies to	Minimum storage volume per VM in GB
		This volume is mounted on /var/log directory.	IG	
diameterapp	diameterapp_data_volume	This volume is used on Diameter App VMs. This volume is mounted on /appdata directory.	ME IG	1
	diameterapp_log_volume	This volume is used on Diameter App VMs. This volume is mounted on /var/log directory.	ME IG	10
iohd	iohd_data_volume	This volume is used on IOHD VMs. This volume is mounted on /appdata directory.	ME IG	1
	iohd_log_volume	This volume is used on IOHD VMs. This volume is mounted on /var/log directory.	ME IG	10
smapp	smapp_data_volume	This volume is used on SM App VMs. This volume is mounted on /appdata directory.	SM IG	1
	smapp_log_volume	This volume is used on SM App VMs. This volume is mounted on /var/log directory.	SM IG	10
	smapp_kafka_volume	This volume is used on SM App VMs. The mountpoint is /kafka_data where Kafka data is stored.	SM	10

Anti-affinity rules

The anti-affinity rules are applied automatically. For more information, contact Nokia Professional Services.

CBAM configuration

CBAM is the only VNF manager that can use the available CSAR packages and Scalable files required for the SPS installation.

Note:

When planning the SPS network, ensure that the communication between CBAM and OAME will be available and the CBAM VM will be able to communicate with both OAME VMs (SSH over IPv4 or IPv6 depending on the network configuration used).

In earlier SPS versions, CBAM had to be configured before installing the SPS. Currently this configuration is not required. Any SPS-specific ansible configurations are automatically bundled into the CSAR package.

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

SPS security hardening

Security hardening attempts to "harden" or "tighten" a system. In general, hardening activities include an operating system, services, middleware, databases, and so on to provide more secure, yet usable systems by minimizing the number of possible attacks. The SELinux security module in the SPS is “enabled” and “permissive”. The SPS comes “pre-hardened” but default passwords must be changed.

For information on passwords and user management, see sections *Security and user management* and *User administration* in *Operation, Administration and Maintenance Guide*.

Chapter 3. SPS installation

This chapter describes how to install the SPS with single and multiple SM, ME and IG sites in the supported OpenStack and VMware cloud environments.

Installation artifacts

Overview

This section describes the artifacts required for the SPS installation in OpenStack and VMware.

Note:

To update the configuration files and generate the VNF artifacts used for the SPS installation, refer to and use the SPS VNF Artifacts Generator Tool Guide. Once the artifacts are generated, see SPS installation *(on page 14)* to proceed with the SPS installation using the cloud platform applicable to your deployment.

The following table lists the SPS installation artifacts.

Table 8. Artifacts

Artifact	OpenStack	VMware
Images	yes	yes
Cloud Service Archive (CSAR) packages	yes	yes
Scalable files	yes	yes
VNF Extension files	yes	yes
Open Virtualization Format (OVF) files	no	yes

The following table lists the images for the SPS installation in OpenStack and VMware.

Table 9. Images

VM name segment	Image name	Applies to
oame	<SPS version>.OAME	SM ME IG
auxiliary	<SPS version>.AUXILIARY	SM ME IG
ioho	<SPS version>.LOADBALANCER_SMIG	SM IG
ioho	<SPS version>.LOADBALANCER_ME	ME
iohd	<SPS version>.DIAMETER_IOH	ME IG
db	<SPS version>.DATABASE	SM ME IG
cdr	<SPS version>.CDR	ME IG
diameterapp	<SPS version>.POLICY_CHARGING_APP	ME IG
comsvc	<SPS version>.COMMON_SERVICES	ME IG
smapp	<SPS version>.SM_APP	SM IG

The installation artifacts are used in accordance with the following:

- CSAR packages, VNF Extension files, Scalable files and images are aligned with each other and must be of the same version.
- CSAR packages, VNF Extension files and Scalable files differ by site type and specific to ME, SM and IG sites.
- Each site requires its own Scalable file and its own VNF Extension file.
- If the same network topology and cloud resources are used for all the ME sites in your deployment, then you can use the same ME CSAR package for all the ME sites, the same SM CSAR package for all the SM sites and the same IG CSAR package for all the IG sites in your deployment.
- MD5 files are available to verify the images and the configuration files. Every image has a corresponding MD5 file and every configuration file has a corresponding MD5 file.

Installing SPS in OpenStack

Overview

Note:

Before installing the SPS, ensure that you have read SPS installation prerequisites *(on page 4)* and completed all the preliminary steps outlined in that chapter.

This section describes the workflow of the SPS installation in OpenStack.

Note:

The minimum SPS deployment is one SM site (VNF) and one ME site (VNF), or one IG site (VNF).

To install the SPS in OpenStack:

1. Contact Nokia Professional services to obtain the following:
   - QCOW2 images
   - CSAR packages
   - Scalable files
   - VNF Extension files
   - MD5 files

   Note:

   See the following:

   - For instructions on how to generate the CSAR packages, the Scalable files and the VNF Extension files, see the SPS VNF Artifacts Generator Tool User Guide.
   - For information on the Scalable files, see the SPS VNF Artifacts Generator Tool User Guide. The Scalable files cannot be manually updated since they are automatically generated.
   - For information on the VNF Extension files, see section VNF Extension file *(on page 36)*.

- For more information on standard lifecycle management and VNF maintenance operations in CBAM such as uploading CSAR packages, creating, modifying and instantiating a VNF, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

2. Download the following artifacts:
   - QCOW2 images
   - CSAR packages
   - Scalable files
   - VNF Extension files
   - MD5 files

3. Using the instructions in Nokia Support Portal, verify the downloaded QCOW2 images and files by generating MD5 hash values and comparing the generated values with the values provided in Support Portal and in MD5 files. The generated MD5 hash values, the MD5 hash values in Support Portal and the values in MD5 files must match. If you find discrepancies, download the QCOW2 images and files again. For more information on the images, see Table 2 *(on page 14)*.

   Note:

   In addition to MD5 values, Nokia provides SHA-256 hash values. The SPS does not use SHA-256 hash values for image verification.

4. Upload to OpenStack the QCOW2 images using the procedure Uploading images to OpenStack *(on page 19)*.

5. Log in to the CBAM GUI and upload the CSAR packages for this deployment using the procedure Uploading a CSAR package to CBAM *(on page 20)*.

6.

   Note:

   All the parameters in the VNF Extension file, including <sdc_config_data> parameter, must be updated before instantiation if you are installing the SPS and before SU if you are upgrading the SPS. If a parameter or a group of parameters can be updated post-instantiate and post-SU, the SPS documentation outlines this information.

   For each site in your deployment, do the following:

   a. In the CBAM GUI, do the following:

      i. Create a VNF with a particular CSAR package using the procedure Creating a VNF *(on page 21)*.
      ii. Modify this VNF with the VNF Extension file generated for this VNF using the procedure Modifying a VNF *(on page 22)*.
      iii. Instantiate this VNF with the Scalable file generated for this VNF using the procedure Instantiating a VNF *(on page 24)*.

   b. Log in to the VNF and change the password using the procedure *(on page)*.
   c. Verify this VNF using the procedure *(on page)*.

d. Optionally, if you choose to use DFSec, do the following procedures:

i. *(on page)*
ii. Enabling Web SSO on SM App VMs *(on page 35)*

For more information about DFSec, see *Overview Guide*.

Installing SPS in VMware

Overview

Note:

Before installing the SPS, ensure that you have read SPS installation prerequisites *(on page 4)* and completed all the preliminary steps outlined in that chapter.

This section describes the workflow of the SPS installation in VMware. The SPS is deployed on one Organization Virtual Data Center (VDC).

Note:

The minimum SPS deployment is one SM site (VNF) and one ME site (VNF), or one IG site (VNF).

To install the SPS in VMware:

1. Contact Nokia Professional services to obtain the following:
   - VMDK images
   - CSAR packages
   - Scalable files
   - VNF Extension files
   - OVF files
   - MD5 files

   Note:

   See the following:

   - For instructions on how to generate the CSAR packages, the Scalable files and the VNF Extension files, see the SPS VNF Artifacts Generator Tool User Guide.
   - For information on the Scalable files, see the SPS VNF Artifacts Generator Tool User Guide. The Scalable files cannot be manually updated since they are automatically generated.
   - For information on the VNF Extension files, see section VNF Extension file *(on page 36)*.
   - For more information on standard lifecycle management and VNF maintenance operations in CBAM such as uploading CSAR packages, creating, modifying and instantiating a VNF, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

2.

Note:

Put the OVF files and the VMDK images into the same specified path on the vCloud VM.

Download the following artifacts:

- VMDK images
- CSAR packages
- Scalable files
- VNF Extension files
- OVF files
- MD5 files

3. Using the instructions in Nokia Support Portal, verify the downloaded VMDK images and files by generating MD5 hash values and comparing the generated values with the values provided in Support Portal and in MD5 files. The generated MD5 hash values, the MD5 hash values in Support Portal and the values in MD5 files must match. If you find discrepancies, download the VMDK images and files again. For more information on the images, see Table 2 *(on page 14)*.

Note:

In addition to MD5 values, Nokia provides SHA-256 hash values. The SPS does not use SHA-256 hash values for image verification.

4. Upload the OVF files and VMDK images to vCloud using the procedure Uploading images and Vapp template to vCloud *(on page 20)*.

5. Log in to the CBAM GUI and upload the CSAR packages for this deployment using the procedure Uploading a CSAR package to CBAM *(on page 20)*.

6.

Note:

All the parameters in the VNF Extension file, including <sdc_config_data> parameter, must be updated before instantiation if you are installing the SPS and before SU if you are upgrading the SPS. If a parameter or a group of parameters can be updated post-instantiate and post-SU, the SPS documentation outlines this information.

For each site in your deployment, do the following:

a. In the CBAM GUI, do the following:

i. Create a VNF with a particular CSAR package using the procedure Creating a VNF *(on page 21)*.
ii. Modify this VNF with the VNF Extension file generated for this VNF using the procedure Modifying a VNF *(on page 22)*.
iii. Instantiate this VNF with the Scalable file generated for this VNF using the procedure Instantiating a VNF *(on page 24)*.

b. Log in to the VNF and change the password using the procedure *(on page)*.
c. Verify this VNF using the procedure *(on page)*.
d. Optionally, if you choose to use DFSec, do the following procedures:

   i. *(on page)*
   ii. Enabling Web SSO on SM App VMs *(on page 35)*

   For more information about DFSec, see *Overview Guide*.

Uploading images to OpenStack

Overview

This procedure provides instructions on how to upload the QCOW2 images to OpenStack. In addition to uploading the QCOW2 images, this procedure provides instructions on how to tune the network performance in OpenStack. This tuning is needed for Nokia internal performance testing and customer deployments and is achieved by enabling the virtio-net multi-queue.

Note:

In OpenStack environmnet, the virtio-net multi-queue allows to scale the network performance with increased number of vCPUs, by letting vCPUs transfer packets through more than one virtqueue pair at a time.

To upload the QCOW2 images to OpenStack:

1. Upload the QCOW2 images to the OpenStack Image service (Glance) and enable the virtio-net multi-queue as follows:
   - During the upload, to each image add the following:

     ```
     --property hw_vif_multiqueue_enabled=true
     ```

     For example,

     ```
     S glance image-create --name <SPS version>.<image name or id> \
     --disk-format qcow2 \
     --file <SPS version>.<image name or id>.qcow2 \
     --container-format bare --property hw_vif_multiqueue_enabled=true
     ```

   For more information on <multiqueue/config> parameter, see *(on page)*.

2. For every SPS flavor modify the OpenStack flavor as follows:

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

Uploading images and Vapp template to vCloud

Overview

This procedure provides instructions on how to upload a Vapp template, which refers to the OVF file and the VMDK images, to vCloud using vCloud Director.

To upload the Vapp template to vCloud:

1. Log in to VMware vCloud Director.
2. In vCloud Director, click the **System** tab, then click **Manage and Monitor**, select **Organizations** and then double click **Organizations name**.
3. If the catalog for the SPS exists, go to the next step. Otherwise, create the catalog for the SPS as follows:
   - Click the **System** tab, select the organization for the SPS, click the **Catalogs** tab. The **My Organization's Catalog** pane appears.
   - In the **My Organization's Catalog** pane, click the **Catalogs** tab and then click **+**. The **New Catalog** window appears.
   - In the **New Catalog** window, enter the name for the catalog and click **Next**. The **Select Storage Type** window appears.
   - In the **Select Storage Type** window, select the **Pre-provision on specific storage policy** button. The **Storage Policy** pane appears.
   - In the **Storage Policy** pane, select the **vSAN Default Storage Policy** and associate it with the OVDC.
4. Upload the OVF file for the SPS OVDC:
   - Click the **System** tab, select the organization for the SPS, click the **Catalogs** tab. The **My Organization's Catalog** pane appears.
   - In the **My Organization's Catalog** pane, click **vApp Templates** and then click the upload button. The **Upload OVF package as a vApp Template** window appears.
   - In the **Upload OVF package as a vApp Template** pane, select **Local file**, click **Browse** and select the OVF file to upload. Click the **Upload** button.
   - Ensure that the OVF file is in the **vApp Templates** list.

   Note:

   The VMDK images are also uploaded in this step by reference in the OVF file.

Uploading a CSAR package to CBAM

Overview

This procedure provides instructions on how to upload a CSAR package to CBAM.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure that you are logged in to the CBAM GUI.

To upload the CSAR package to CBAM:

1. In the CBAM GUI, click the options button with three horizontal lines and then select **Catalog**. The **Catalog** pane appears.
2. In the **Catalog** pane, click the **Upload VNF Package** represented by the plus icon. The **Upload VNF Package** window appears.
3. In the **Upload VNF Package** window, click **browse** and select the appropriate CSAR package, or use **Drag and drop** option to add the CSAR package to the CBAM catalog.
4.

   Note:

   The CSAR package is ready for upload when the following occurs in the **Upload VNF Package** window:

   - The name of the CSAR package appears in the **File name** area.
   - In the **Status** field, the message `Ready for upload` appears.

   When the CSAR package is ready for upload, click **Upload**.

   Expected outcome

   If the operation succeeds, the CSAR package appears in the **Catalog** pane. If the operation fails, an error message appears in the GUI.
5. If the operation succeeds, verify that the CSAR package is listed in the CBAM catalog.
6. If the operation fails, depending on the cause of failure, you might need to delete the CSAR package using the procedure Deleting a CSAR package *(on page 34)*. For more information on what to do if this operation fails, see CloudBand Application Manager (CBAM) documentation.

Creating a VNF

Overview

This procedure provides instructions on how to create a VNF in CBAM.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure that you are logged in to the CBAM GUI.

To create the VNF:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Catalog**. The **Catalog** pane appears.
2. In the **Catalog** pane, select the appropriate CSAR package and hover the mouse over the right of the selected line, and then click the **Create VNF from Package** button represented by the multiple processes icon. The **Create New VNF** window appears.
3. In the **Create New VNF** window, do the following:
   - In the **Name** field, enter the name for this VNF.
   - If needed, in the **Description** field, provide additional information about this VNF.
   - Click **Create**.

   Expected outcome

   If the operation succeeds, the new VNF appears in the **Virtualized Network Functions** pane with the **Last Operation Status** marked as `Completed - Create`. If the operation fails, an error message appears in the GUI.
4. If the operation fails, depending on the cause of failure, you might need to delete the VNF. For example, if you used a wrong CSAR package, then delete the VNF using the procedure Deleting a VNF *(on page 33)*. For more information on what to do if this operation fails, see CloudBand Application Manager (CBAM) documentation.

Modifying a VNF

Overview

This procedure is a VNF maintenance operation that provides instructions on how to change in the CBAM GUI the VNF's extensions modifiable attribute by uploading the VNF Extension file prepared for this VNF.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure the following:

- You are logged in to the CBAM GUI.
- The VNF Extension file has been prepared for this VNF. For more information, see section VNF Extension file *(on page 36)*.

Note:

All the parameters in the VNF Extension file, including <sdc_config_data> parameter, must be updated before instantiation if you are installing the SPS and before SU if you are upgrading the SPS. If a parameter or a group of parameters can be updated post-instantiate and post-SU, the SPS documentation outlines this information.

To modify the VNF using the VNF Extension file:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Virtualized Network Functions**. The **Virtualized Network Functions** pane appears.

2. In the **Virtualized Network Functions** pane, select the appropriate VNF and hover the mouse over the right of the selected line, and then click the **Modify** button represented by the pencil icon. The **General Data** pane appears.

3. In the **General Data** pane, to upload the VNF Extension file, click **Yes** to **Do you want to upload the operation parameters as a file attachment?** and then click **Continue**. The **Parameters** pane appears.

4. In the **Parameters** pane, click **browse** and select the appropriate VNF Extension file, or use **Drag and drop** option to upload the file.

   Note:

   Each VNF requires its own VNF Extension file.

   Note:

   The VNF Extension file is ready for upload when the following occurs in the **Parameters** pane:

   - The name of the VNF Extension file appears in the **File name** area.
   - In the **Status** field, the message `Ready for upload` appears.

5. When the VNF Extension file is ready for upload, click **Finish**.

   Expected outcome

   If the operation succeeds, the **Virtualized Network Functions** pane appears where this VNF is listed with the **Last Operation Status** marked as `Completed - Modify Info`. If the operation fails, an error message appears in the GUI.

6. If the operation fails, depending on the cause of failure, you might need to repeat this procedure. For example, if you used a wrong VNF Extension file or the VNF Extension file has errors. For more information on what to do if this operation fails, see CloudBand Application Manager (CBAM) documentation.

Instantiating a VNF

Overview

This procedure provides instructions on how to instantiate a VNF in CBAM.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Note:

Ensure that all the virtual IP addresses used for the SPS deployment are excluded from the allocation pool of your cloud environment, OpenStack or VMware.

Before beginning this procedure, ensure the following:

- You are logged in to the CBAM GUI.
- The Scalable file for this VNF is ready for uploading to CBAM. For more information, see Scalable file *(on page 36)*.

To instantiate the VNF:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Virtualized Network Functions**. The **Virtualized Network Functions** pane with the list of created VNFs appears.
2. In the **Virtualized Network Functions** pane, select the appropriate VNF and hover the mouse over the right of the selected line, and then click the **Instantiate** button represented by the start icon. The **General Data** pane appears.
3. In the **General Data** pane, do the following:
   - In the **Instantiation level id** field, select the value from the drop-down list.
   - In the **VIM password** field, enter the password for the VIM access.
   - Click **Continue**. The **Parameters** pane appears.
4. In the **Parameters** pane, click **browse** and select the appropriate Scalable file, or use **Drag and drop** option to upload the Scalable file for this VNF.

   Note:

   Each VNF requires its own Scalable file.

   Note:

   The Scalable file is ready for upload when the following occurs in the **Parameters** pane:

- The name of the Scalable file appears in the **File name** area.
- In the **Status** field, the message `Ready for upload` appears.

5. When the Scalable file is ready for upload, click **Finish** to automatically use the Scalable file and start this VNF's instantiation.

   Note:

   If you click **Cancel**, the **Exit Form?** window appears, and then if in the **Exit Form?** window, you click **Cancel**, the **Parameters** pane appears, if you click **Exit**, the **Virtualized Network Functions** appears.

   Expected outcome

   The **Virtualized Network Functions** pane appears where the **Last Operation Status** of this VNF shows as `Processing - Instantiate`. If this VNF is instantiated successfully, the **Last Operation Status** of this VNF changes to `Completed - Instantiate`.

   Note:

   The instantiation operation takes at least 40 minutes.

6. If the instantiation fails, your further actions depend on the cause of failure and might be one of the following:
   - If the instantiation fails due to cloud platform issues and those issues are resolved, then instantiate this VNF again using this procedure.
   - If the instantiation fails due to CBAM issues, then refer to CloudBand Application Manager (CBAM) documentation for troubleshooting and once the issues are resolved, instantiate this VNF again using this procedure.
   - If the instantiation fails due to incorrect configurations, then fix the errors and proceed accordingly with the appropriate procedures. See Additional procedures *(on page 32)* for any other procedures that you might need such as Terminating a VNF *(on page 32)* or Deleting a VNF *(on page 33)*.

   If needed, contact Nokia Professional Services.

Chapter 4. SPS upgrade

This chapter describes how to upgrade the SPS deployed in OpenStack or VMware from one version to the other version. In this chapter the SPS version that you are upgrading from is called FROM load version, and the SPS version that you are upgrading to is called TO load version.

Upgrading a VNF

Overview

This procedure provides instructions on how to upgrade a VNF of any type.

Note:

During the SU the following might occur:

- If an error occurs during the upgrade that requires you to back out of the SU, contact Nokia Professional Services to determine the backout strategy.
- A switchover might cause the traffic to go down on IOHO and IOHD VMs.

Note:

If you need to back out, use the procedure *(on page)*.

To upgrade the VNF:

1. If this VNF is an ME or IG site, set the ME version parameter in the SM GUI to the FROM load value using the procedure Updating ME version in SM GUI *(on page 30)*.
2. Prepare pAutoSU using the procedure: *(on page)*.
3. Generate the TaskList file using the procedure *(on page)*.
4. Check the health of the SPS by using the procedure *(on page)*.
5. Execute the pAutoSU script. See Executing pAutoSU *(on page 27)* for information about how to execute the pAutoSU script.

Downloading a file with instantiated VNF info

Overview

This procedure provides instructions on how to assign the instantiated VNF’s information to a specified parameter.

Before you begin:

Before beginning this procedure, ensure that you are logged in to the CBAM GUI.

To assign the VNF information:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Virtualized Network Functions**. The **Virtualized Network Functions** pane with the list of created VNFs appears.
2. In the **Virtualized Network Functions** pane, click the appropriate VNF. The **Virtualized Network Function Details** window appears.
3. In the **Virtualized Network Function Details** window, click the **Instantidated VNF Info** tab.

   The **Instantidated VNF Info** pane appears.
4. In the **Instantidated VNF Info** pane, click the **Download** button represented by the downstream arrow icon.

   Expected outcome

   The file `vnf-<VNF name>-instantiated-info.json` is downloaded.
5. Save the file `vnf-<VNF name>-instantiated-info.json` to the **$AUTO_HOME** directory.

Executing pAutoSU

Overview

This procedure provides instructions on how to execute the pAutoSU script such that it performs the upgrade automatically.

The SPS ISU will upgrade VMs within a VNF, one group at a time, for a total of three groups using the pAutoSU script. When the script runs for each group, there are multiple tasks that are completed by the script. The pAutoSU script triggers the ISU using a command line that can be used with an option (-p or --pause) that causes a pause after each task is complete within a single group. That is, the script will pause and wait for the user's input before continuing to the next task. When the script pauses, you can answer "c" , for continue, at the prompt. You can forego the pause and continue steps using the "r" option after the pause at any task, which allows the ISU to continue to the last task without anymore pauses. Other options also exist that allow you to specify the tasklist file (<task_list_file>) or the VNF Instance (-i <vnf_instance_id>) on which to execute the script.

Note:

You should have already generated the TaskList file. If you have not already generated the Tasklist file, generate the TaskList file using the procedure *(on page)*. The TaskList file defines tasks to be executed in one run and in sequence.

For information about TaskList, see section TaskList file *(on page 63)*.

To execute pAutoSU:

1.

Note:

During this step, there might be a need to perform the procedure *(on page)*. For information about TaskList, see TaskList file *(on page 63)*.

As `root` user, execute pAutoSU as follows:

a. Perform one of the following as required:

- For VMware, enter:

```
cd $AUTO_HOME
```

- For OpenStack, enter:

```
cd $AUTO_HOME
```

```
source <PROJECT>-openrc.sh
```

Based on the version of OpenStack, you may be prompted to OS_PASSWORD when you source the openrc file. At the prompt enter your OS_PASSWORD if required:

```
Please enter your OpenStack Password for project <PROJECT> as user
<OS_USERNAME>: <OS_PASSWORD>
```

2. Enter one of the following:
   - To execute the pAutoSU with a specific task list and pause after executing each task successfully: `./autoSU.py -f <task_list_file> [-p|--pause]`
   - Run with specific task list:

```
./autoSU.py -f <task_list_file>
```

   - Run with specific task list and a specific VNF instance id:

```
./autoSU.py -f <task_list_file> -i <vnfid>
```

   - Allow pAutoSU to continue tasks from wherever it stopped and exited:

```
./autoSU.py -f <task_list_file> -c
```

Where:

-h or **--help** prints this help message and exits.

"**-p**" or "**--pause**" is an optional parameter that allows a pause after executing each task successfully. You will need to enter "c" to continue to execute the next task, or press "r" to run through to the last task without a pause, or press "q" to quit.

"**-c**" is an optional parameter used in a new command that allows the pAutoSU script to continue performing tasks from where it stopped and exited.

"**-f <task_list_file>**" is the file which defines the task list (mandatory). The path is tools/genTasklist/tasklist-upgrade.txt, which is for updating all VMs in one attempt, where autoSU.py

is only run the one time. If you choose to upgrade using the before and after tasklist, it means the autoSU.py script is run twice as follows:

- Once for tasklist-upgrade-before-soak.txt—for updating half of VMs in one attempt
- Once for tasklist-upgrade-after-soak.txt—for updating the rest of the VMs

"**-i <vnfid>**" is an optional parameter that allows you to execute the script on a specific VNF instance. If this option is not provided, vnfId value from su_config.yaml will be used.

Expected outcome

The upgrade will proceed to completion. However, if the pAutoSU script fails during a group upgrade, the script will stop and exit. In this case, you will need to run the backout procedure and restart the ISU procedures. See *(on page)* for basic instruction about backing out of an upgrade.

If there is a problem with the upgrade at this point, you will need to troubleshoot the problem as follows:

- The SU_VM $AUTO_HOME/su.log should be examined.
- The CBAM /var/log/cbam/workflows.log should be examined.
- The CBAM /var/log/cbam/<VNF ID>.log should be examined. Where <VNF ID> is the VNF ID of the VNF being upgraded. For example: /var/log/cbam/vnf_CBAM-b9651edb444b4b87ac4dda723652094a.log

Note:

After the upgrade is committed, you must update the version on the non-impacted nodes. This is performed in step 19 of the primary procedure (To upgrade the SPS).

Updating persisted version

Overview

Note:

If you are using DFSEC, ensure you have enabled DFSEC using Enabling DFSEC *(on page 31)* before performing this procedure. By Default DFSEC is disabled in this release.

This procedure provides instructions on how to update the persisted version to a specified load value.

To update the persisted version to a specified load value:

1. Log in to the Active OAME VM as `tpaadmin`.
2. Enter the following to specify the load value:

   ```
   /opt/tpa/bin/sdc-set-persisted-version <IBUILD value>
   ```

   For example,

   ```
   /opt/tpa/bin/sdc-set-persisted-version SPS_20_0_I99
   ```

where SPS_20_0_I99 is the IBUILD value of the TO load.

3. Enter:

   ```
   /opt/tpa/bin/persist-version.sh
   ```

Updating ME version in SM GUI

Overview

This procedure provides instructions on how to update the ME version parameter for an ME or IG site in the SM GUI.

To update the ME version in the SM GUI:

1. As smadmin user, log in to the SM GUI using https://<SM site's IOHO floating IP>:8443/servicemanager/ URL.
2. In the SM GUI, click the **Global Configuration tab** and then click the **Managed Element List** tab. The **Managed Element List** pane appears.
3. In the **Managed Element List** pane, select the appropriate VNF, then click the **Actions** tab and then click **Change the ME Version**. The **Change the ME Version** window appears.
4. From the **New Version** drop-down list, select the appropriate version and click **Submit**.
5. In the SM GUI, verify that the ME version has been updated as expected.

Setting the enabledurabledelete value

Perform this procedure to set the value of **enabledurabledelete** to *false*. This must be done on all ME/IG VNFs.

1. Log into the active OAME as tpaadmin.
2. Enter the following command to change the value:

   sdcctl set system config/tpaproperties/tpapps/db/tementity/enabledurabledelete false

3. Enter the following command to verify the value:
   sdcctl ls --recursive --printvalue /|grep enabledurabledelete
   /<vfn_name>/config/tpaproperties/tpapps/db/tementity/enabledurabledelete false

Enabling DFSEC

Overview

This procedure provides instructions on how to enable DFSEC functionality.

Note:

If you have, or are planning to have an SPS GEO deployment DFSEC enabled (through upgrade or fresh installation), you must not create, update, or delete the SM User/Role and its association until DFSEC is enabled on both sites. For more information, contact Nokia Support.

Chapter 5. Additional procedures

This chapter describes additional procedures that might be used during installation.

Terminating a VNF

Overview

This procedure provides instructions on how to terminate a VNF in CBAM. When a VNF is terminated, it is removed from the VIM. At the same time, this VNF remains in the CBAM VNF catalogue and can be instantiated again.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure that you are logged in to the CBAM GUI.

To terminate the VNF:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Virtualized Network Functions**. The **Virtualized Network Functions** pane with the list of created VNFs appears.

2. In the **Virtualized Network Functions** pane, select the VNF to terminate and hover the mouse over the right of the selected line, and then click the **Terminate** button represented by the stop icon. The **Termination type** pane appears.

3. In the **Termination type** pane, do the following:
   - If this VNF's instantiation was successful, in the **Termination type** field, select **Graceful**.
   - If this VNF's instantiation was not successful, in the **Termination type** field, select **Forceful**.
   - Click **Finish**.

   Expected outcome

   The **Virtualized Network Functions** pane appears where this VNF is listed with the **Last Operation Status** marked as `Processing - Terminate`. If the termination succeeds, the **Last Operation Status** changes to `Complete - Terminate`.

4. If the termination fails, do the following:
   - If the termination fails for the first time, repeat this procedure but choose **Forceful** as the termination type.

- If the termination fails again, contact the cloud administrator to investigate the cause of failure and clean up the cloud platform before executing this procedure again using **Forceful** termination type.

Deleting a VNF

Overview

This procedure provides instructions on how to delete a VNF in CBAM.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure that you are logged in to the CBAM GUI.

To delete the VNF:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Virtualized Network Functions**. The **Virtualized Network Functions** pane with the list of created VNFs appears.
2. In the **Virtualized Network Functions** pane, select the VNF to terminate and hover the mouse over the right of the selected line, and then click the **Delete** button represented by the trash icon. The **Delete VNF?** window appears.
3. Click **OK**.

   Expected outcome

   The **Virtualized Network Functions** pane appears without this VNF.

Changing CSAR package version

Overview

This procedure provides instructions on how to change the version of a CSAR package for a particular VNF in CBAM.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure the following:

- You are logged in to the CBAM GUI.
- The CSAR packages are uploaded to CBAM.

To change the version of the CSAR package:

1. In the CBAM GUI, click the option button with three horizontal lines, and then select **Virtualized Network Functions**. The **Virtualized Network Functions** pane with the list of created VNFs appears.
2. In the **Virtualized Network Functions** pane, select the appropriate VNF and hover the mouse over the right of this VNF, then click the **More** button represented by the icon with three vertical dots and then click **Change Package Version**. The **Select VNF package Id** pane appears.
3. In the **Select VNF package Id** pane, select the appropriate version of the CSAR package and click **Continue**. The **Parameters** pane appears.
4. In the **Parameters** pane, if needed, upload a new Scalable file and then click **Finish**.
5. Verify that the VNF's VNFD ID has been changed.

Deleting a CSAR package

Overview

This procedure provides instructions on how to delete a CSAR package from CBAM.

Note:

For more information on CBAM, see CloudBand Application Manager (CBAM) documentation in the Nokia Discovery Center by going to https://doc.networks.nokia.com

Before you begin:

Before beginning this procedure, ensure that you are logged in to the CBAM GUI.

To delete the CSAR package from CBAM:

1. In the CBAM GUI, click the options button with three horizontal lines, and then select **Catalog**. The **Catalog** pane appears.
2. In the **Catalog** pane, click select the appropriate CSAR package and hover the mouse over the right of the selected line, and then click the **Delete Package** button represented by the trash icon. The **Delete VNF package?** window appears.
3. Click **OK**.

Expected outcome

The **Catalog** pane appears without this CSAR package.

Enabling Web SSO on SM App VMs

Overview

This procedure provides instructions on how to enable Web SSO on SM App VMs.

To enable DFSec on Web SSO on SM App VMs:

1. Log in to any SM App VM as tpaadmin user.

   Note:

   Before repeating this procedure for the next SM App VM, ensure that jersey is UP as follows:

   a. Tail the logs using `tail -f /opt/tpa/logs/SPSServer.log  | grep -i jersey` command.
   b. Wait for `Jersey is now Ready Starting Web Traffic now` line to appear in the logs.

2. Stop the TPA server using `TPA_control stop` command.

3. Go to the path `/opt/tpa/osgi/instance`

4. Execute:

   ```
   mv  keycloak.cfg.disabled  keycloak.cfg
   ```

5. In `/opt/tpa/osgi/instance/webagent.override.descriptor.properties` file, do the following:
   - Find the following line:

     ```
     "com.nokia.tpapps.servicemanager-gui:/opt/SPS_20_3_R1/osgi/instance/
     keycloak/keycloak.gui.web.xml"
     ```
   - If the line is present and commented, uncomment the line.

     Expected outcome

     ```
     com.nokia.tpapps.servicemanager-gui:/opt/SPS_20_3_R1/osgi/instance/
     keycloak/keycloak.gui.web.xml
     ```
   - If the line is absent, add at the end of the file the following:

     ```
     com.nokia.tpapps.servicemanager-gui:/opt/SPS_20_3_R1/osgi/instance/
     keycloak/keycloak.gui.web.xml
     ```

6. Start the TPA server using `TPA_contol start` command.

7. Repeat this procedure for each SM App VM.

Chapter 6. Reference material

This chapter describes the files that are used in the SPS installation and the SPS upgrade.

Scalable file

Overview

The Scalable file is used by CBAM during VNF instantiation. The Scalable file is generated for OpenStack and VMware deployments. The Scalable files cannot be manually updated since they are automatically generated. For information on the Scalable files, see the SPS VNF Artifacts Generator Tool Guide.

VNF Extension file

Overview

The VNF Extension file is automatically generated with the SPS VNF Generator Tool. However, certain parameters in the VNF Extension file can be manually updated. This section describes the VNF Extension parameters, including <sdc_config_data> (service discovery and configuration) parameter, and provides information on how to modify the parameters that must or might be updated.

Note:

The great majority of the parameters in the VNF Extension file apply to all the site types: SM, ME and IG. However, certain parameters are applicable only to particular site types.

Modify vnf extension file

Note:

- All the parameters in the VNF Extension file, including <sdc_config_data> parameter, must be updated before instantiation if you are installing the SPS and before SU if you are upgrading the SPS. If a parameter or a group of parameters can be updated post-instantiate and post-SU, the SPS documentation outlines this information.
- You must modify <sdc_config_data> parameter for each site in the deployment.
- You must contact Nokia Professional Services to modify <sdc_config_data> parameter.
- Do not remove VNF Extension parameters and parameters defined in <sdc_config_data> parameter.
- All the parameters in <sdc_config_data> parameter that represent internal and external VIPs must be aligned with the values provided for the CSAR packages and the Scalable files.

To modify the VNF Extension file:

1. Modify the VNF Extension parameters as needed and required. For more information, see section *(on page)*.
2. Modify <sdc_config_data> parameter as needed and required. For more information, see the following sections:
   - BR section *(on page 37)*
   - Bootstrap ETCD section *(on page 43)*
   - Bootstrap MariaDB *(on page 46)*
   - CDR section *(on page 47)*
   - *(on page)*
   - *(on page)*
   - Fluentd section *(on page 54)*
   - *(on page)*
   - HA section *(on page 55)*
   - HTTPD section *(on page 55)*
   - IPConfig section *(on page 55)*
   - Kafka section *(on page 56)*
   - *(on page)*
   - *(on page)*
   - *(on page)*
   - ZooKeeper section *(on page 61)*

BR section

The BR section is used for the backup and restore settings related to configuring backup repositories and adding the Veritas NetBackup server.

The following table describes the parameters of the BR section.

Note:

Do not remove the parameters.

Note:

Ensure that there is sufficient disk space where you intend to store the backup files including the oame_backup_volume. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.

Table 10. BR parameters descriptions

Parameter	Description	Update notes
BR/backup_ip_type	This parameter indicates which IP type is used for the backup and restore.	Modify if needed.

Parameter	Description	Update notes
	The following values are supported: • IPv4 • IPv6 This parameter must be set. It cannot be the empty string.	
BR/backup_mode	This parameter indicates what backup mode is used for the backup and restore. The following values are supported: • NONE • NETBKUP where: • NONE indicates that the backup and restore is local. • NETBKUP is the backup and restore with the NetBackup server. If you choose NETBKUP, you must install the client.	Modify if needed.
BR/net_interface	This parameter indicates which net interface is used for the backup and restore. The following values are supported: • external • internal If the NetBackup server is used, <BR/net_interface> must be set to *external* because the backup server should be in a different data center than the SPS.	Modify if needed.
BR/package_url	This parameter defines the URL link that is used to download the NetBackup client. The value is http://1.1.1.1/download/NetBackup_CLIENT.tar.gz This parameter must be set if you want to use HTTP to retrieve the software. This parameter must be set to the empty string if <BR/backup_mode> is NetBackup, and you want to use FTP to retrieve the software. In this case, the FTP package parameters must be set.	Modify if needed.

Parameter	Description	Update notes
	This parameter can be set to the empty string if <BR/backup_mode> is NONE. In this case, any value assigned to this parameter is ignored.	
BR/partition_type	This parameter indicates which backup repository is used. The values are as follows: • CV • NFS where: • CV is Cinder volume for OpenStack and Independent disk for VMware. • NFS is NFS export.	Modify if needed.
BR/ sched_bkup_precedence	This parameter defines the precedence of one group of VMs over the other group of VMs in scheduled backup. This parameter applies only to group backups with dependencies. The possible values are: • ["vdu-OAME", "vdu-DB"] • empty string (“ “) This parameter must be set to ["vdu-OAME", "vdu-DB"] . When the SPS is backed up, vdu-OAME must always be backed up before vdu-DB. If this parameter is set to the empty string, then no precedence is needed. If a VDU group specified in the value does not exist, backup runs as scheduled.	Do not modify.
BR/package_ftp BR/package_ftp parameters define the NetBackup FTP settings. The package FTP parameters are ignored if the following applies: • <BR/backup_mode> is NONE. • <BR/package_url> is configured, If <BR/backup_mode> is NETBKUP, the package FTP parameters must be configured.		
ftp_path	This parameter defines the path which is used to download the client package using FTP. For example:	Modify if needed.

Parameter	Description	Update notes
	• ftp://1.1.1.1/download/NetBackup_CLIENT.tar.gz is the FTP path for the NetBackup client.	
key	This parameter defines the key in the FTP package. The following values are supported: • NETBKUP	Modify if needed.
passwd	This parameter defines the password which is used to access the NetBackup clients using FTP.	Modify if needed.
user_name	This parameter defines the user name which is used to access the NetBackup clients using FTP.	Modify if needed.
BR/partition_info		
BR/partition_info parameters define the backup repository settings.		
IP	The values are as follows: • If <BR/partition_info/key> is CV, the value is the empty string. • If <BR/partition_info/key> is NFS, this parameter must be set and the value is the IP address of the remote NFS server.	Modify if needed.
key	This parameter defines the key. The values are as follows: • CV • NFS where: • CV is Cinder volume. • NFS is the network file system.	Modify if needed.
path	The values are as follows: • If <BR/partition_info/key> is CV, the value is the empty string. • If <BR/partition_info/key> is NFS, this parameter must be set.	Modify if needed.
tag	This parameter defines the disk volume. For the OpenStack, it defines the Cinder volume. For VMware, it defines the Independent disk. The values are as follows: • If <BR/partition_info/key> is CV, this parameter must be set to the following value: ◦ for VMware, *sdc*	Modify if needed.

Parameter	Description	Update notes
	◦ for OpenStack *vdc* • If <BR/partition_info/key> is NFS, the value is the empty string.	
BR/server_info BR/server_info parameters define the NetBackup remote backup server settings. These parameters should be configured when the <BR/backup_mode> is NETBKUP. If <BR/backup_mode> is NONE, the server_info parameters are ignored.		
host_IP	This parameter defines the IP address of the remote backup server.	Modify if needed.
host_name	This parameter defines the host name of the remote backup server.	Modify if needed.
key	This parameter defines the remote backup server. The following values are supported: • NETBKUP	Modify if needed.

The following figure shows an example of the BR section. In this example, the backup and recovery is configured to use external network interface and NetBackup as a backup server. The client software in this example is retrieved using HTTP.

Figure 2. BR section configured with external network and NetBackup server

```
"BR": {
 "[Global]": {},
 "[Local]": {},
 "[System]": {
  "BR/backup_ip_type": "IPv4",
  "BR/backup_mode": "NETBKUP",
  "BR/net_interface": "external",
  "BR/package_ftp": [
   {
    "ftp_path": "ftp://1.1.1.1/download/NetBackup_CLIENT.tar.gz",
    "key": "NETBKUP",
    "passwd": "appplat",
    "user_name": "appplat"
   },
   {
    "ftp_path": "ftp://1.1.1.1/download/NetWorker_CLIENT.rpm",
    "key": "NETWK",
    "passwd": "appplat",
    "user_name": "appplat"
   }
  ],
```

```
  "BR/package_url": "http://111.111.11.111/dev_umage/
NetBackup_7.7.3_CLIENTS2.tar.gz",
  "BR/partition_info": [
   {
    "IP": "",
    "key": "CV",
    "path": "",
    "tag": "vdc"
   },
   {
    "IP": "1.1.1.1",
    "key": "NFS",
    "path": "/home",
    "tag": ""
   }
  ],
  "BR/partition_type": "CV",
  "BR/sched_bkup_precedence": [
   "vdu-OAME",
   "vdu-DB"
  ],
  "BR/server_info": [
  {
   "host_IP": "11.111.1.11",
   "host_name": "netbackup",
   "key": "NETBKUP"
  },
  {
   "host_IP": "1.1.1.1",
   "host_name": "networker",
   "key": "NETWK"
   }
  ]
}
}
```

Figure 3. BR configured with NFS repository

The following example shows the BR section configured with NFS as a backup repository.

```
"BR": {
  "[Global]": {},
  "[Local]": {},
  "[System]": {
    "BR/backup_ip_type": "IPv4",
    "BR/backup_mode": "NONE",
    "BR/net_interface": "internal",
    "BR/package_ftp": [
      {
        "ftp_path": "ftp://1.1.1.1/download/NetBackup_CLIENT.tar.gz",
        "key": "NETBKUP",
        "passwd": "appplat",
        "user_name": "appplat"
```

```
      },
      {
        "ftp_path": "ftp://1.1.1.1/download/NetWorker_CLIENT.rpm",
        "key": "NETWK",
        "passwd": "appplat",
        "user_name": "appplat"
      }
    ],
    "BR/package_url": "http://1.1.1.1/download/NetBackup_CLIENT.tar.gz",
    "BR/partition_info": [
      {
        "IP": "",
        "key": "CV",
        "path": "",
        "tag": "sdc"
      },
      {
        "IP": "135.252.170.46",
        "key": "NFS",
        "path": "/home/NFS",
        "tag": ""
      }
    ],
    "BR/partition_type": "NFS",
    "BR/BR/sched_bkup_precedence": ["vdu-OAME","vdu-DB"],
    "BR/server_info": [
      {
        "host_IP": "1.1.1.1",
        "host_name": "netbackup",
        "key": "NETBKUP"
      },
      {
        "host_IP": "1.1.1.1",
        "host_name": "networker",
        "key": "NETWK"
      }
    ]
  }
}
```

Bootstrap ETCD section

The Bootstrap ETCD section contains the parameters required for the installation of the ETCD cluster. In the SPS, ETCD is configured as Local ETCD where peer and client addresses must be listed. These addresses define the internal IP addresses of the Auxiliary VMs that act as ETCD servers.

Bootstrap ETCD parameters

The following table describes the parameters of the Bootstrap ETCD section.

Note:

The following applies to the Bootstrap ETCD parameters:

- Do not remove the parameters.
- If you have changed the internal network host identifier, you must ensure that this change is consistent with all the applicable IP address values of sdc_config_data parameter.

Table 11. Bootstrap ETCD parameters descriptions

Parameter	Description	Update notes
Confd Backend Type	This parameter defines the backend type used for the ETCD cluster. Supported values are: etcd for v2 of ETCD, etcdv3 for v3 of ETCD, and sdc for the versioning data feature.	Do not modify.
Data Dir	This parameter defines the directory to store ETCD data, that is, wal and snapshot files. This parameter represents the environment variable ETCD_DATA_DIR.	Do not modify.
Election Timeout	This is one of the ETCD cluster's time parameters used by the underlying distributed consensus protocol. This parameter defines time (in milliseconds) for an election to timeout. The timeout is the interval that a follower node will go without hearing a heartbeat before attempting to become leader itself. This parameter should be based on the heartbeat interval and average round-trip time between members. Election timeouts must be at least 10 times greater than the round-trip time so it can account for variance in the network. The upper limit is 50000 ms, which should only be used when deploying a globally-distributed ETCD cluster. The election timeout value should be the same for all the members in one cluster. This parameter represents the environment variable ETCD_ELECTION_TIMEOUT.	Contact Nokia Professional Services to change the default value if needed.
Etcd CA File	This is the file name of the certificate used for SSL/TLS connections to ETCD. When this option is set, you can set advertise-client-urls using HTTPS schema. This parameter represents the environment variable ETCD_CA_FILE.	N/A
Etcd Cert File	Etcd Cert File and Etcd Key File are the signed key pair. This parameter represents the environment variable ETCD_CERT_FILE.	N/A
Etcd Client Cert Auth	This parameter defines whether client cert authentication is enabled or disabled. Supported values are "FALSE" for disabled, and "TRUE" for enabled, that is, provide the client key and cert file. This parameter represents the environment variable ETCD_CLIENT_CERT_AUTH.	N/A

Parameter	Description	Update notes
Etcd Key File	Etcd Cert File and Etcd Key File are the signed key pair. This parameter represents the environment variable ETCD_KEY_FILE.	N/A
Etcdctl Cert File	This is the file name of the client cert file. It is required only if Etcd Client Cert Auth is set to TRUE. This parameter represents the environment variable ETCDCTL_CERT_FILE.	N/A
Etcdctl key file	This is the file name of the client key file. It is required only if Etcd Client Cert Auth is set to TRUE. This parameter represents the environment variable ETCDCTL_KEY_FILE.	N/A
Heartbeat Interval	This is one of the ETCD cluster's time parameters used by the underlying distributed consensus protocol. This parameter defines the time (in milliseconds) of a heartbeat interval. This is the frequency with which the leader will notify followers that it is still the leader. The heartbeat interval value is recommended to be around the maximum of the average round-trip time (RTT) between members, normally around 0.5-1.5x of the round-trip time. The easiest way to measure the RTT is to use the PING utility. The heartbeat interval value should be the same for all the members in one cluster. This parameter represents the environment variable ETCD_HEARTBEAT_INTERVAL.	Contact Nokia Professional Services to change the default value if needed.
Network Config	The value is set to *Local*.	Do not modify.
Remote Servers	This parameter represents remote ETCD servers. It is required only if Network Config is set to Remote, which the SPS does not use. The value is set to empty square brakets "[]".	Do not modify.
Snapshot Count	This parameter defines the number of ETCD changes before a snapshot is done. If the use of ETCD’s memory and disk is too high, you can lower the snapshot threshold. The values such as 5000 or 1000 are acceptable.	Contact Nokia Professional Services to change the default value if needed.
System Name	This parameter represent the system name of a VNF. The SPS does not use this parameter. The value is the empty string.	Do not modify.
Local ETCD This parameter defines the peer and client addresses of the Auxiliary VMs acting as servers.		
Client Address	This parameter is the internal IP address of the Auxiliary VM.	Must update, if you have changed the internal network host identifier.
Peer Address	This parameter is the internal IP address of the Auxiliary VM.	

Parameter	Description	Update notes
		Must ensure that this change is consistent with all the applicable IP address values of sdc_config_data parameter.

Bootstrap MariaDB

The Bootstrap MariaDB section is used for configuring parameters for Bootstrap MariaDB component. The following table describes the parameters of the Bootstrap MariaDB section.

Note:

Do not remove the parameters.

Table 12. Bootstrap MariaDB parameters descriptions

Parameter	Description	Update notes
Add Users		
This parameter is a collection of records of MariaDB users that must be added to the database at installation time. This parameter contains two predefined users.		
Grant Object	This parameter defines the DB object to which a user is given access.	Do not modify.
Grant Privilege	This parameter defines whether or not a user is granted any privileges.	Do not modify.
Host	This parameter defines the host. If this parameter is left blank, than the user is given access from any host.	Do not modify.
Password	This parameter defines user password.	Do not modify.
User	This parameter defines username.	Do not modify.
User *alma*	This parameter defines the database user for alarms.	Do not modify.
User *sa*	This parameter defines the database user for Web SSO service.	Do not modify.
Built-in Users		
This parameter is a collection of records of built-in users for MariaDB. Each record contains a password and a username.		
User *root*	This parameter defines MariaDB root user. The password for this user applies to all root users for all hosts during database installation. The root user password must be the same for all the hosts in a cluster or a replication set.	Do not modify the username and the password.
User *repl@b.c*	This parameter defines the default replication user, which is assigned in the database during installation in a replication configuration.	Do not modify the username

Parameter	Description	Update notes
		and the password.
User *maxscale*	This parameter defines the user for MaxScale monitoring and management task for a database cluster.	Do not modify the username and the password.

CDR section

The CDR section is used for the settings related to charging data records (CDRs).

The following table describes the CDR parameters.

Note:

Do not remove the parameters.

Table 13. CDR parameters descriptions

Parameter	Description	Update notes
cdr/AuditCDR	This parameter defines how the CDR application is audited; whether or not and how the file is deleted from the CDR VM. This parameter is defined as follows: `{"enabled":"yes","age":7,"time_started":"0000"}` where: • <enabled> specifies whether the CDR audit is enabled or not • <age> is the age of the CDR files in days. The value range is [1,180]. • <time_started> is the time when the audit shall start. The format of this value is hhmm, for example, 2359, which is one minute to midnight.	N/A
cdr/timestampInMS	If this parameter is true, then the **RecordTimestamp** tag in CDR/EDR generated is in millisecond format with timezone offset. Since **RecordTimestamp** tag is generated in GMT time, timezone offset will be +0000.	To change the value of timestampInMS flag at run-time (after deployment or upgrade), contact Nokia Support.
cdr/enableSFTP	This parameter defines whether the third parties such as the billing system, analytics or customer support collect CDRs using SFTP or do not collect.	N/A

Parameter	Description	Update notes
	The values are true or false. If this parameter is true, then the third parties collect CDRs. The values of true and false are case sensitive and must be in lowercase.	
cdr/global_properties/ log.connection.close	This parameter defines whether the log broker disconnects or not. It can be turned off when interacting with 0.9 brokers with an aggressive <connection.max.idle.ms> value. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
cdr/global_properties/ message.send.max.retries	This parameter defines the number of times the producer retries sending a failing MessageSet. Retrying might cause reordering. The value range is [0,10000000].	N/A
cdr/global_properties/ message.timeout.ms	This parameter defines the local message timeout. This value is only enforced locally and limits the time that a produced message waits for a successful delivery. The time of 0 (zero) indicates infinite. The value range is [0,900000].	N/A
cdr/monitoring/sleep-interval	This parameter defines how often the CDR service is checked by the monitoring service. The value is in seconds.	N/A
cdr/producer/encoding/ common/cdr.data.list.size	This parameter defines the number of CDRs in a buffered list before being pushed to messaging queue. This can be changed in run-time and does not require service restart. The value range is [5,20].	N/A
cdr/producer/ encoding/common/ cdr.encoding.thread.pool.size	This parameter defines the number of threads in the executor service working on encoding CDR data. This can be configured in run-time and does not require service restart. The value range is [5,20].	N/A
cdr/producer/encoding/ common/cdr.header.param	This parameter defines the version of 3GPP that is specified in the CDR header and according to 3GPP 32.297 document. This parameter is an array that contains the following 3GPP secifications: • ReleaseId The value range is [1,6]. For the values of 7 and above, ReleaseIdExt comes into picture. The default value is 7. • Version The value range is [0,32]. The default value is 2. • TSnumber	N/A

Parameter	Description	Update notes
	The value range is [1,7] and [9,20]. The default value is 7. • ReleaseIdExt The value range is [0,3]. The default value is 0. For more information, see section 6.1.2 (CDR file format and transfer document) of 3GPP 32.297 document.	
cdr/producer/ kafka_producer/acks	This parameter defines the number of acknowledgments that the producer requires the leader to receive before considering the request completed. This controls the durability of records that are sent. The supported values are *all*, 0 (zero) and 1. • If the value is 0 (zero), then the producer does not wait for any acknowledgment from the server at all. The record is immediately added to the socket buffer and considered sent. In this case, no guarantee can be made that the server has received the record, and the retries configuration does not take effect since Client generally is not aware of any failures. The offset given back for each record is always set to -1. • If the value is 1, the leader writes the record to its local log but responds without waiting for the full acknowledgement from all the followers. In this case, if the leader fails immediately after acknowledging the record but before the followers have replicated this record, then this record is lost. • If the value is *all*, the leader waits for the full set of in-sync replicas to acknowledge the record. This guarantees that the record will not be lost as long as at least one in-sync replica remains alive. This is the strongest available guarantee.	N/A
cdr/producer/ kafka_producer/batch.size	This parameter defines the default batch size in bytes. The producer attempts to batch records together into fewer requests whenever multiple records are sent to the same partition. This improves performance on the client and on the server. No attempt will be made to batch records larger than this size. The requests sent to brokers contain multiple batches, one for each partition with data available to be sent. A small batch size makes batching less common and may reduce throughput. (The zero batch size disables batching entirely.) A very large batch size may use memory a bit more	N/A

Parameter	Description	Update notes
	wastefully as we always allocate a buffer of the specified batch size in anticipation of additional records.	
cdr/producer/ kafka_producer/ bootstrap.servers	This parameter is a list of host/port pairs for establishing the initial connection to the Kafka cluster. The client uses all the servers irrespective of which servers are specified for bootstrapping. This list only impacts the initial hosts used to discover the full set of servers.	N/A
cdr/producer/ kafka_producer/ buffer.memory	This parameter defines the total number of memory bytes that the producer can use to buffer records waiting to be sent to the server. If the records are sent faster than they can be delivered to the server, the producer blocks for <max.block.ms> after which it throws an exception. This setting should correspond roughly to the total memory that the producer is going to use, but this is not a hard bound since not all this memory is used for buffering. Some additional memory is used for compression, if compression is enabled, as well as for maintaining in-flight requests. The value range is 0...	N/A
cdr/producer/ kafka_producer/linger.ms	This parameter adds a small amount of artificial delay. Rather than sending out a record immediately, the producer waits for up to the specified delay to allow other records to be sent so that these records can be batched together. The value range is 0...	N/A
cdr/producer/ kafka_producer/ max.block.ms	This parameter controls for how long CDR sending is blocked either because the buffer is full or metadata is unavailable. Blocking the user-supplied serializers or partitioner is not counted against this timeout. The value range is 0...	N/A
cdr/producer/ kafka_producer_mgr/ sendpool.size	This parameter defines the number of threads used by the client to send message to Kafka. The value range is 1...	N/A
cdr/producer/ kafka_producer/ request.timeout.ms	This parameter defines the maximum amount of time that the client waits for the response of a request. If the response is not received before the timeout elapses, the client resends the request if necessary or fails the request if retries are exhausted. The value range is 0...	N/A
cdr/producer/ kafka_producer/retries	Setting a value greater than zero causes the client to resend any record whose send fails with a potentially transient error. This is similar to when the client resends the record upon receiving an error.	N/A

Parameter	Description	Update notes
	Allowing retries without setting <max.in.flight.requests.per.connection> to 1 potentially changes the ordering of records because if two batches are sent to a single partition and the first one fails and after failure it is resent, but the second batch succeeds, then the records from the second batch may appear first. The value range is [0,2147483647].	
cdr/topics	This parameter defines the topic name, its partition count and Group ID as follows: `[{\"Name\":\"cdr\",\"Partition\":\"12\", \"GroupId\":\"1\"}`	N/A
cdr/topic_properties/ produce.offset.report	This parameter reports the offset of a produced message back to the application. The application must use the dr_msg_cb to retrieve the offset from rd_kafka_message_t.offset. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
cdr/topic_properties/ request.required.acks	This parameter indicates how many acknowledgements the leader broker must receive from ISR brokers before responding to the request. • If the value is 0 (zero), the leader broker does not send any response/ack to the client. • If the value is 1, only the leader broker needs to acknowledge the message. • If the value is -1 or *all*, the leader broker blocks until the message is committed by all in sync replicas (ISRs) or broker's <in.sync.replicassetting> before sending a response. The value range is [-1,1000].	N/A
cdr/restartTime	This parameter defines the sleep time in minutes before running `systemctl restart cdr` for the CDR VM. This parameter is used during scaling operations.	Default values shall only be modified if you need to increase the restart time between CDR VMs during scaling.
cdr/restartBufferTime	This parameter defines the time buffer in minutes between two restart-trigger of CDR VMs. This parameter is used during scaling operations.	Default values shall only be modified if you

Parameter	Description	Update notes
		need to increase the restart time between CDR VMs during scaling.
CDR varTable parameters The optional *cdr/varTable/<key name>* parameters define the CDR *vars* environment where <key name> is an arbitrary string.		
cdr/varTable/var3	The CDR daemon automatically rejoins the cluster to continue receiving CDR messages and writing CDR files whenever disk usage percentage drops below the low threshold. The value format is as follows: `CDR_DISK_THRESHOLD_LOW=90`	N/A
cdr/varTable/var4	The CDR daemon audits the disk usage percentage periodically. Once CDR disk usage percentage is over the high threshold, the CDR daemon automatically leaves from the CDR cluster with an alarm. The CDR that detaches itself from the cluster does not receive a CDR message and does not write CDR files. The value format is as follows: `CDR_DISK_THRESHOLD_HIGH=97`	N/A
cdr/varTable/var5	CDR_COMMIT_TIME = *n* means that CDR commits message every *n* seconds. A value of 0 (zero) indicates that CDR is committed as soon as it is received. The value format is as follows: `CDR_COMMIT_TIME=5`	N/A
CDR index parameters CDR index parameters contain <group index> portion, which is a number in the range of [0,200]. However, the usage of the value zero (“0”) is restricted by the SPS platform, thus the value zero cannot be used. The group index is used to bifurcate data based on the following: • CDR or EDR • record type • MVNO implementation Currently only indexes 1 and 2 are used to distinguish CDRs and EDRs.		

Parameter	Description	Update notes
cdr/<group index>/dest_dir	This parameter defines the path where the CDR file that corresponds to the group index is written. Ensure that there is sufficient disk space for cdr_record_volume where the CDR/EDR files are stored. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set the appropriate size.	N/A
cdr/<group index>/ file_header_info	This parameter defines the following: • Routing filter that determines the routing of CDRs into the CDR file. • Private extension that indicates the vendor-specific private extension to be included into the CDR file. • Node IP address. • Wether of not there is a release extension. This part indicates the 3GPP release of TS 32.298. The values are Y and N. Beyond Rel-9, the value should be Y.	N/A
cdr/<group index>/ file_name_info	This parameter defines the following: • whether or not the local time is used • file name format • file suffix. The maximum length is 21. The file name format is a string that can be composed of any printable character which is valid for the file name. “$$” is “$” in the file name. The CDR is defined with the following variables: • YEAR is a 4-digit year. • MONTH is a 2-digit month in the range of [01, 12]. • DAY is a 2-digit day in the range of [01, 31]. • JULIAN is a 3-digit Julian date in the range of [001, 366]. • HOUR is a 2-digit hour in the range of [00, 24]. • MINUTE is a 2-digit minute in the range of [00, 59]. • SECOND is a 2-digit second in the range of [00, 59]. • RC is a running counter value. For example, `MAS021_-_${RC}.${YEAR}${MONTH}${DAY}_-_ ${HOUR}${MINUTE}`	N/A

Parameter	Description	Update notes
cdr/<group index>/ file_size_limit	This parameter specifies the CDR file size limit. The value range is from 1 to 1000 MB.	N/A
cdr/<group index>/ index_name	This parameter defines the index name for the group index.	N/A
cdr/index_list	This parameter defines the list of index.	N/A
cdr/<group index>/node_ID	This parameter defines the system that is used in the naming of the CDR files. The value is an alphanumeric (digits and numbers only) name from 1 to 12 characters.	N/A
cdr/<group index>/rc_info	This parameter defines the following: • RC file size. • whether or not the RC field of a CDR file name needs to be padded.	N/A
cdr/<group index>/ rec_count_limit	This parameter specifies the number of records in a CDR file. The value range is from 100 to 1,000,000.	N/A
cdr/<group index>/ time_interval	This parameter specifies the time interval in minutes of how often a CDR file is created. The options are 1, 5, 10, 15, 30 and 60 minutes.	N/A

Fluentd section

This section is used for the settings related to Fluentd. The following table describes the parameters of the Fluentd section.

Note:

When updating this section:

- Do not remove the parameters.
- Do not modify any ports.
- If you have changed the internal network host identifier, you must ensure that this change is consistent with all the applicable IP address values of sdc_config_data parameter.

Table 14. Fluentd parameters descriptions

Parameter	Description	Update notes
fluentd/app_mappings	This parameter defines mapping rules for all the VM other than OAME.	N/A
fluentd/app_template	This parameter defines Fluentd configuration template for all the VM other than OAME.	N/A
fluentd/app_values	This parameter defines Fluentd configuration values. By default this parameter is set to the OAME internal IP addresses.	Must update with customer values.
fluentd/ fluentd_backup_dirs	This parameter defines Fluentd backup directory. This parameter must be set to `/etc/td-agent/plugin`.	Do not modify.

Parameter	Description	Update notes
fluentd/ fluentd_oam_group_name	This parameter defines OAME group name.	N/A
fluentd/plugin_port	This parameter defines the list of ports used by Fluentd plugin.	Do not modify.
fluentd/oame_mappings	This parameter defines mapping rules for OAME VMs.	N/A
fluentd/oame_template	This parameter defines configuration template for OAME VMs.	N/A
fluentd/oame_values	This parameter defines Fluentd configuration values. By default this parameter is set to the OAME internal IP addresses.	Must update with customer values.
fluentd/out_forward_port	This parameter defines Fluentd out-forward port. The default port is 24224.	Do not modify.
fluentd/fluentd_port	This parameter defines Fluentd port. The default port is 8514.	Do not modify.

HA section

The HA section is used to define the HA configuration of the OAME, IOHO and IOHD (ME only) pairs. For more information on the HA section, see the VNF Artifacts Generator Tool Guide.

The HA mechanism relies on keepalived, which relies on VRRP. A VRRP instance is defined for each VM pair (OAME, IOHO and IOHD) and for the internal interface only. The internal virtual IP (VIP) addresses are managed by the HA mechanism.

Note:

The entire HA section and the parameters comprised in it are automatically created when the VNF extension file is generated. This section does not exist in the sample VNF Extension files available for the SPS release. Do not remove or manually modify the generated parameters. Any changes needed for the HA section must be made as documented in the VNF Artifacts Generator Tool Guide.

HTTPD section

The HTTPD section is used for the Apache HTTP Server (“httpd”) and its web server settings. It is only used on the Active OAME VM.

Note:

Do not remove or modify the parameters.

IPConfig section

The IPConfig section is related to the LCM framework and is used for configuring IP settings for VNFs and various VM types. For more information on the IPConfig section, see the VNF Artifacts Generator Tool Guide.

Note:

The entire IPCONFIG section and the parameters comprised in it are automatically created when the VNF Extension file is generated. This section does not exist in the sample VNF Extension files available for the SPS release. Do not remove or manually modify the generated parameters. Any changes needed for the IPConfig section must be made as documented in the VNF Artifacts Generator Tool Guide.

Kafka section

The Kafka section is used for the settings related to Kafka. The following table describes the parameters of the Kafka section.

Note:

Do not remove the parameters.

Table 15. Kafka parameters descriptions

Parameter	Description	Update notes
kafka/kafka_reserved_broker_max_id	This parameter defines the maximum number that can be used for a broker ID.	N/A
kafka/kafka_delete_topic_enable	This is a switch to enable topic deletion. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
kafka/kafka_rack_enable	This is a switch to enable Kafka rack-awareness. Although the supported values are true and false, this parameter must always be set to true.	Do not modify.
kafka/kafka_broker_port	This parameter defines the port that the socket server listens to.	Do not modify.
kafka/kafka_num_network_threads	This parameter defines the number of threads handling network requests.	N/A
kafka/kafka_num_io_threads	This parameter defines the number of threads handling disk I/O.	N/A
kafka/kafka_socket_send_buffer_bytes	This parameter defines the size of the send buffer used by the socket server.	N/A
kafka/kafka_socket_receive_buffer_bytes	This parameter defines the size of the receive buffer used by the socket server.	N/A
kafka/kafka_socket_request_max_bytes	This parameter defines the maximum size of a request that the socket server accepts (protection against OOM).	N/A
kafka/kafka_datavolume	This is a comma separated list of directories for storing log files.	N/A

Parameter	Description	Update notes
	Ensure that there is sufficient disk space for comsvc_kafka_volume and smapp_kafka_valume where Kafka data is stored. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.	
kafka/kafka_num_partitions	This parameter defines the number of log partitions per topic. More partitions allow greater parallelism for consumption, but this results in more files across the brokers.	N/A
kafka/ kafka_num_recovery_threads_per_data_dir	This parameter defines the number of threads per data directory for log recovery at startup and flushing at shutdown. It is recommended to increase this value for installations with data dirs located in RAID array.	N/A
kafka/kafka_log_retention_hours	This parameter defines the minimum age of a log file to be eligible for deletion.	N/A
kafka/kafka_log_retention_minutes	This parameter defines the number of minutes to keep a log file before deleting it. If this is not set, the value in <kafka/ kafka_log_retention_hours> is used.	N/A
kafka/kafka_log_retention_bytes	This parameter controls the maximum size that a partition, which consists of log segments, can grow to before we discard old log segments to free space. Deletion always starts from the end of a log.	N/A
kafka/kafka_log_segment_bytes	This parameter defines the maximum size of a log segment file. When this size is reached, a new log segment is created.	N/A
kafka/ kafka_log_retention_check_interval_ms	This parameter defines the interval for checking log segments whether these log segments should be deleted or not according to the retention criteria.	N/A
kafka/kafka_zk_port	This is the ZooKeeper port that Kafka connects to.	Do not modify.

Parameter	Description	Update notes
kafka/kafka_zk_connection_timeout_ms	This parameter defines the timeout in ms for connecting to ZooKeeper.	N/A
kafka/kafka_message_max_bytes	This parameter defines the largest record batch size allowed by Kafka. If this size is increased and there are consumers older than 0.10.2, then these consumers' fetch size must also be increased. In the latest message format version, records are grouped into batches for efficiency. In previous message format versions, uncompressed records are not grouped into batches This limit only applies to a single record.	N/A
kafka/kafka_replica_fetch_max_bytes	This parameter defines the number of bytes of messages to attempt to fetch for each partition. This is not the absolute maximum. If the first record batch in the first non-empty partition of the fetch is larger than this value, the record batch is still returned to ensure that progress can be made.	N/A
kafka/kafka_default_replication_factor	This parameter defines the default replication factors for automatically created topics.	Do not modify.
kafka/ kafka_offsets_topic_replication_factor	This parameter defines the replication factor for the offsets topic. This is set higher to ensure availability. Internal topic creation fails until the cluster size meets this replication factor requirement.	N/A
kafka/kafka_log_dirs	This is the debug log directory for Kafka. Ensure that there is sufficient disk space for comsvc_log_volume and smapp_log_volume where the log is stored. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.	N/A
kafka/kafka_update_configuration	This parameter indicates whether Kafka configuration can be updated by “update configuration” LCM.	N/A

Parameter	Description	Update notes
	The values are true or false. The values of true and false are case sensitive and must be in lowercase.	
kafka/kafka_log_save_enable	This parameter defines whether to back up and recover a log file or not for SPS upgrade and backout. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
kafka/kafka_mount_disk	This parameter defines whether to mount external storage for log files or not. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
kafka/kafka_heap	This is a JVM parameter to start Kafka.	N/A
kafka/kafka_broker_groupname	This parameter identifies Kafka broker name which is required for scaling operations. For ME and IG, the value is vdu-COMSVC, For ME, the value is vdu-SMApp.	N/A
kafka/ kafka_reassign_partitions_throttle_Bps	This defines the bandwidth assigned with the partition reassignment operation. This partition reassignment operation is triggered when the VMs that host Kafka server are scaled-in or scaled-out. Nokia recommends to set this parameter to 50% of the total bandwidth between the two VMs hosting Kafka server. The higher the bandwidth allocation is, the faster the reassignment operation completes.	To modify the default value, use configure-kafka-partition-keys tool. For information on the tool, see the Operation, Administration and Maintenance Guide.
kafka/ kafka_enable_scale_reassign_partition	This parameter defines whether or not during a scaling operation kafka_reassign_partition tool executes automatically. However, the values are true or false, Nokia recommends to set this parameter to true.	Do not modify.

Parameter	Description	Update notes
	If it occurs that this parameter is set to false, use configure-kafka-partition-keys tool to set this parameter to true. For information on the tool, see the Operation, Administration and Maintenance Guide.	
kafka/kafka_repartition_delay_second	This parameter defines how often to check whether or not a Kafka scaling (scale-out or scale-in) operation has completed.	Do not modify.
kafka/kafka_repartition_retries_count	This parameter defines the maximum time for Kafka scale-out and scale-in operations. If a scaling operation finishes before this time, (a check is done after every <kafka_repartition_delay_second>), then it will end the operation . If it does not complete in time, then kafka partition reassignment is still running. In this case, verify the reassignment script to complete before running any other action on the node" log will be printed. Manually, we have to check status of partition before executing any other operation.	Do not modify.
kafka/kafka_to_load_rpm_version	This parameter is used if Kafka is upgraded. In case of the upgrade operation where FROM_load version is greater than TO_load version, this parameter has to be set to rpm version of Kafka corresponding to TO_load before performing upgrade as well as in the sdc_update_conf of TO_load. For example, if there is an upgrade from 2.0.1(FROM_load kafka version) to 1.1.0 (TO_load kafka version), then this parameter must be set to "1.1.0" before performing upgrade (i.e. in 2.0.1 load) as well as in the sdc_update_conf of 1.1.0 load.	Do not modify.
kafka/ backout_incompatible_kafka_version	This parameter is used if Kafka is upgraded. Once Kafka version is upgraded to 2.1.x load, it is not possible to downgrade Kafka. A generic provision is provided by setting "kafka/	Modify if needed.

Parameter	Description	Update notes
	backout_incompatible_kafka_version" to "2.1" as the default value. If a later version becomes also downgrade incompatible, then this parameter has to be set to that later Kafka version before performing kafka upgrade as well as in the sdc_update_conf of new load.	

ZooKeeper section

This section is used for the settings related to ZooKeeper.

The following table describes the parameters of the ZooKeeper section.

Note:

Do not remove the parameters.

Table 16. ZooKeeper parameters descriptions

Parameter	Description	Update notes
zookeeper/ zookeeper_group_name	This is the group name for ZooKeeper.	Do not modify.
zookeeper/ zookeeper_datavolume	This is a comma separated list of directories for storing log files. Ensure that there is sufficient disk space for auxiliary_data_volume where the log files are stored. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.	N/A
zookeeper/zookeeper_log_dirs	This is the debug log directory for Zookeeper. Ensure that there is sufficient disk space for auxiliary_log_volume where the directory is stored. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.	N/A
zookeeper/ zookeeper_data_subdir	This is the sub-directory for ZooKeeper data under zookeeper_datavolume. Ensure that there is sufficient disk space for auxiliary_data_volume where the sub-directory is stored. All VM sizing is done prior to instantiation and there is no	N/A

Parameter	Description	Update notes
	re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.	
zookeeper/ zookeeper_datalog_subdir	This is the sub-directory for ZooKeeper transaction logs under zookeeper_datavolume. Ensure that there is sufficient disk space for auxiliary_data_volume where the sub-directory is stored. All VM sizing is done prior to instantiation and there is no re-sizing. Prior to instantiation, see the SPS Dimensioning and Engineering Guidelines for dimensioning information and the SPS VNF Artifacts Generator Tool Guide for instructions on how to set an appropriate size.	N/A
zookeeper/zookeeper_tick_time	This parameter defines the number of milliseconds of each tick.	N/A
zookeeper/zookeeper_init_limit	This parameter defines the number of ticks that the initial synchronization phase can take.	N/A
zookeeper/ zookeeper_sync_limit	This parameter defines the number of ticks that can pass between sending a request and getting an acknowledgement.	N/A
zookeeper/ zookeeper_client_port	This parameter defines the port for connecting with clients.	N/A
zookeeper/zookeeper_peer_port	This parameter defines the port for connecting with peers.	N/A
zookeeper/ zookeeper_election_port	This parameter is the election port for ZooKeeper.	N/A
zookeeper/ zookeeper_max_client_cnxns	This parameter defines the maximum number of client connections. To handle more clients, the value might be increased.	N/A
zookeeper/ zookeeper_update_configuration	This parameter indicates whether ZooKeeper configuration can be updated by “update configuration” LCM. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
zookeeper/ zookeeper_log_save_enable	This parameter defines whether to back up and recover a log file or not for SPS upgrade and backout. The values are true or false. The values of true and false are case sensitive and must be in lowercase.	N/A
zookeeper/ zookeeper_mount_disk	This parameter defines whether to mount external storage for log files or not. The values are true or false.	N/A

Parameter	Description	Update notes
	The values of true and false are case sensitive and must be in lowercase.	
zookeeper/zookeeper_heap	This is a JVM parameter to start ZooKeeper.	N/A
zookeeper/ zookeeper_auto_purge_interval	This parameter defines the time interval in hours for which the purge task has to be triggered. Setting this parameter to a positive integer (1 and above) enables autopurging. Setting this parameter to 0 disables autopurging.	N/A
zookeeper/ zookeeper_snap_retain_count	This parameter defines autopurging behavior in ZooKeeper. This parameter defines how many of the most recent snapshots in <zookeeper_data_subdir> and how many of the most recent transaction logs in <zookeeper_datalog_subdir> are retained. The minimum value is 3.	N/A

TaskList file

Overview

This section describes the TaskList file used in the SU.

Syntax

The syntax in the TaskList file is as follows:

```
task = command timeout node {parameters}
command = upgrade/backout/commit/setdbversion/updatevnfinfo/
changevnfdversion
node = VDU-type node-index
VDU-type = OAM/IOHO/IOHD/DiameterAPP/COMSVC/Auxiliary/DB/SMAPP
node-index = 1*3 DIGIT; 0 to 255
parameters = {*parameter
parameter = pname[:pvalue]
group-tasks = {1* upgrade-task}/{1*backout-task}/{1*commit-task}
upgrade-task = task; command = upgrade
backout-task = task; command = backout
commit-task  = task; command = commit
```

Group SU

The SPS SU supports upgrade/backout/commit of a group of VMs in parallel. To support the zone-based parallel SU, a group can be one of the following:

- VMs with the same availability zone
- subset of a particular availability zone because of CBAM limited capabilities

The following is an example of the group SU:

```
{
upgrade    OAM       0
upgrade    Auxiliary  2
upgrade    DB       0
upgrade    Auxiliary  3
upgrade    DB       2
}
```

Upgrade task

The upgrade task upgrades a VM to the TO load version. This task consists of the following steps:

1. switchover: to check the HA state of the HA VMs (OAME or IOHO or IOHD) to be upgraded; if the HA state is ACTIVE, then switchover is performed to switch to STANDBY state.
2. change package version: to change to the interim SU package.
3. prepare: to invoke the prepared hook functions of each component on the VM to back up the data and shut down running services gracefully.
4. change package version: to change to the TO load package.
5. update stack: to update the stack using HEAT with the new HOT and stack parameters. HEAT rebuilds the VM with the specified target Load image and starts the VM with cloud-init.
6. apply: to invoke the apply hook functions for each component on the VM to deploy the and start component service. Then pAutoSU tool will wait here to let operator perform post-install steps on the VM.
7. SPS start service: to confirm the SPS service launch after the postinstall steps, requires an operator's interaction.
8. switchover: to switchover the upgraded HA VMs (OAME or IOHO or IOHD) to Active, only for upgrade group2.

The following is an example of the upgrade task:

```
{
upgrade   OAM   0
}
```

Backout task

The backout task rolls the upgraded VM back to the latest running point. This task consists of the following steps:

1. switchover: to check the HA state of the HA VMs (OAME or IOHO or IOHD) to be backed out; if the HA state is ACTIVE, then switchover is performed to switch to STANDBY state.
2. change package version: to change to the TO load package.
3. prepare: to invoke the hook functions of each component on the VM to shut down gracefully running services.
4. change package version: to change to the interim SU package.

5. update stack: to update the stack using HEAT with the original HOT and stack parameters. HEAT rebuilds the VM with the from load image and starts the VM with cloud-init.
6. apply: to invoke the apply hook functions for each component on the VM to launch the component service and clean up the SU state and backup data.
7. startService: to invoke the start service hook functions for each component on the VM to start the component service. Then pAutoSU tool will wait here to let operator perform post-install steps on the VM.
8. SPS start service: need operator's interaction to confirm launch the SPS service after the post-install steps.
9. switchover: only for backout group3, to perform switchover to make the backed out HA VMs (OAME or IOHO or IOHD) ACTIVE.

The following is an example of the backout task:

```
{
backout   OAM   0
}
```

Commit task

The commit task is invoked after the VM has been upgraded successfully. This task consists of the following steps:

1. commit: to invoke the hook functions of each component to clean up the SU state and backup data.
2. update stack: update the stack parameters to align with the target load.

The following is an example of the commit task:

```
{
commit   OAM   0
}
```

Setdbversion task

The SPS provides a DB wrapper layer to access DB with a configured DB schema. The DB persisted version is set to the “FROM Load” value before the SU. The SU allows the upgraded service to read/write data from/to DB using the “FROM Load” schema, and allows DB replication across SPS sites with the same schema.

After all the SPS sites have been upgraded, the DB persisted version is changed to “TO Load”. Then the SPS service will access the DB with “TO Load” schema and if the existing data is “FROM Load”, it will convert the data and write it using the “TO Load” schema.

The setdbversion task sets the DB persisted version in the SDC, and takes effect on the running SPS VMs.

The following is an example of the setdbversion task, and the value of "LoadVersion" should be statically set to the "fromload":

```
setdbversion  OAM   0   {"LoadVersion":"fromload"}
```

Configuration update

The SPS configuration data is defined in JSON format as an element of the VNF Extensions structure. The VNF Extensions should be modified with the updated configuration data in SU preparation. The configuration data is applied into SDC in the upgrade phase with an explicit indication.

The configuration update is not a standalone task but a parameter of the first upgrade task defined as follows:

"loadNewConfiguration":"yes"

The following is an example of how the configuration update is used:

```
{
upgrade   Auxiliary   2   {"loadNewConfiguration":"yes"}
}
```

Change VNFD version

This task changes the VNFD Descriptor for the current VNF. The value is assigned in the "vnfdVersion" parameter with the configuration name in `su_config.yaml` file.

The following is an example of the change VNFD version task:

```
changevnfdversion NA  NA  {"customFromVNFDVersion"}
```

Update VNF info

This task modifies the current VNF info using the VNF Extension file.

The following is an example of the update VNF info task, and the value of "vnfd_extension_file" should be statically set to "toVNFDExtension" or "fromVNFDExtension".

```
updatevnfinfo  NA  NA  {"vnfd_extension_file":"toVNFDExtension"}
```

Upgrade order

In order to speed up the SU, zone-based image SU is introduced. This means that the upgrade order should be the zone upgrade order.

There are three zones supported in this release, zone-3 includes one of the Auxiliary VMs, and it is the first zone to be upgraded. Zone-1 and zone-2 include half of the left SPS VMs. The second upgrade zone is the one that includes the most standby VMs of OAME, IOHO and IOHD. The left zone is the third upgrading zone.

The following is an example of the upgrade order:

```
changevnfdversion NA    NA  {"customFromVNFDVersion"}
updatevnfinfo    NA     NA   {"vnfd_extension_file":"toVNFDExtension"}
```

```
setdbversion    OAM   0  {"LoadVersion":"fromload"}
@groupname=group1
{
upgrade               Auxiliary       2   {"loadNewConfiguration":"yes"}
}
@groupname=group2
{
    upgrade           Auxiliary       0
    upgrade           Auxiliary       3
    upgrade              CDR          0
    upgrade             COMSVC        0
    upgrade               DB          0
    upgrade               DB          2
    upgrade          DiameterAPP      0
    upgrade              IOHD         0
    upgrade              IOHO         0
    upgrade              OAM          0
}
@groupname=group3
{
    upgrade           Auxiliary       1
    upgrade           Auxiliary       4
    upgrade              CDR          1
    upgrade             COMSVC        1
    upgrade               DB          1
    upgrade               DB          3
    upgrade          DiameterAPP      1
    upgrade              IOHD         1
    upgrade              IOHO         1
    upgrade              OAM          1
}
```

Commit order

The commit does not impact the service. This means that all the SPS VMs can commit together.

The following is an example of the commit order:

```
{
    commit        Auxiliary       2
    commit        Auxiliary       0
    commit        Auxiliary       3
    commit           CDR          0
    commit          COMSVC        0
    commit            DB          0
    commit            DB          2
    commit       DiameterAPP      0
    commit           IOHD         0
    commit           IOHO         0
    commit           OAM          0
    commit        Auxiliary       1
    commit        Auxiliary       4
    commit           CDR          1
```

```
    commit           COMSVC        1
    commit             DB          1
    commit             DB          3
    commit        DiameterAPP      1
    commit            IOHD         1
    commit            IOHO         1
    commit            OAM          1
}
```

Backout order

The backout order for the VMs and zones must be the reverse of the upgrade order. If the rollback of any Active/Standby VM creates the first instance running the old software, it must be made Active immediately after the rollback.

In the backout tasklist file, after all the backout tasks, add a changevnfdversion task with <vnfdVersion> set to the VNFD version of the FROM load.