ERICSSON

ENM Configuration System Administrator Guide

Operating Instructions

1/1543-AOM 901 151-1 Uen CJ

Contents

1 ENM Configuration System Administrator Guide

This section describes the system administration tasks for ENM Configuration applications.

Target Group

System Administrator

2 Connect to a Service

2.1 Connect to a Virtual Machine on a Physical ENM Deployment

Prerequisites

A command window is open and you have `superuser` privileges.

Steps

1. Log on to the ENM MS as **`litp-admin`** user and switch to the **`root`** user.
2. List the contents of the host file to view all connected VMs within the deployment.

```
[root@ms-1 ~]# cat /etc/hosts
...
192.168.99.20 svc-1-pmserv # Created by LITP. Please do not edit
192.168.99.26 svc-1-netex # Created by LITP. Please do not edit
192.168.99.16 svc-1-ebc # Created by LITP. Please do not edit
192.168.99.36 svc-1-mspm # Created by LITP. Please do not edit
192.168.99.28 svc-1-uiserv # Created by LITP. Please do not edit
192.168.99.14 svc-1-supervc # Created by LITP. Please do not edit
..
192.168.99.32 svc-1-mscm # Created by LITP. Please do not edit
..
192.168.99.50 svc-1-jms # Created by LITP. Please do not edit
..
192.168.99.3 logstash # Created by LITP. Please do not edit
..
192.168.99.2 httpd # Created by LITP. Please do not edit
192.168.99.40 sso # Created by LITP. Please do not edit
..
192.168.99.12 svc-1-medrout # Created by LITP. Please do not edit
192.168.99.22 svc-1-cmserv # Created by LITP. Please do not edit
192.168.99.52 svc-1-sec # Created by LITP. Please do not edit
192.168.99.8 openidm # Created by LITP. Please do not edit
```

   The aliases for the parallel VMs take the form of `<SVC host>-<service>`.

   For example: `svc-1-cmserv, svc-2-cmserv`.

   The active-passive VMs take the form of `<service>`.

   For example: `httpd, sso, openidm`.

3. To access the VM, copy the private key of the cloud-user from its secure location to the `MS` or `SVC` node.

```
[root@ms-1 ~]# /root/.ssh/vm_private_key
```

Refer to *VM Security Tasks* in the *ENM System Administrator Guide* to learn more about the `vm_private_key`.

4. Connect by `SSH` to the VM you want.

   To access the `VM`, use the `cloud-user` user ID and include the path to the `VM` private key. For example:

```
[root@ms-1 ~]# ssh -i /root/.ssh/vm_private_key cloud-user@svc-1-cmserv
Last login: Thu Feb 26 10:14:43 2015 from 192.110.0.59
[cloud-user@svc-1-cmserv ~]# sudo su - root
[root@svc-1-cmserv ~]#
```

2.1.1 Connect to each ENM Physical Node

Prerequisites

— The root password was changed during the installation process and must be known by the system administrator. This must be repeated on all newly deployed ENM nodes.

— A command window is open.

Steps

1. Log on to each physical node from the MS

```
[root@ms-1 ~]$ ssh litp-admin@<node_hostname>
litp-admin@<node_hostname>'s password:
Last login: Mon Feb 23 11:25:13 2015 from ms-1
[litp-admin@<node_hostname> ~]$ su - root
Password:
[root@<node_hostname> ~]#
```

Note: Once connected, after the initial deployment, the passwords for both the `litp-admin` and `root` users must be changed.

2.2 Connect to a Virtual Machine on an ENM on Cloud Deployment

Prerequisites

— A command window is open and you have `superuser` privileges.

— You have access to the private key file for authentication, contact your OpenStack administrator

Steps

1. List the virtual machine aliases from the consul service:

   Using the private key for authentication, copy the key to the EMP server. Log on to EMP server and list the consul members to view all connected VMs within the deployment:

```
> scp -i <cloud-user private key> <cloud-user private key> cloud-user@<EMP I →
P Address>:/var/tmp/vm_private_key
> ssh -i <cloud-user private key> cloud-user@<EMP IP Address>
[cloud-user@ostk003-emp-0 ~]$ chmod 700 /var/tmp/vm_private_key
[cloud-user@ostk003-emp-0 ~]$ sudo su -
[root@ostk003-emp-0 ~]# consul members
Node                       Address          Status  Type    Build  Protocol  →
 DC
haproxy                    10.3.2.31:8301   alive   client  0.8.1  2         →
 dc1
opendj-1                   10.3.2.83:8301   alive   client  0.8.1  2         →
 dc1
opendj-2                   10.3.2.84:8301   alive   client  0.8.1  2         →
 dc1
openidm                    10.3.2.85:8301   alive   client  0.8.1  2         →
 dc1
ostk003-accesscontrol-0    10.3.1.251:8301  alive   client  0.8.1  2         →
 dc1
ostk003-accesscontrol-1    10.3.1.252:8301  alive   client  0.8.1  2         →
 dc1
ostk003-elasticsearch-0    10.3.2.15:8301   alive   client  0.8.1  2         →
 dc1
...
ostk003-neo4j-2            10.3.2.77:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfscommon-0        10.3.0.81:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfsnrbk-0          10.3.0.83:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfspm-0            10.3.0.85:8301   alive   client  0.8.1  2         →
 dc1
ostk003-nfspm-1            10.3.0.82:8301   alive   client  0.8.1  2         →
 dc1
...
ostk003-secserv-1          10.3.2.98:8301   alive   client  0.8.1  2         →
 dc1
ostk003-serviceregistry-0  10.3.2.100:8301  alive   server  0.8.1  2         →
 dc1
ostk003-serviceregistry-1  10.3.2.101:8301  alive   server  0.8.1  2         →
 dc1
ostk003-serviceregistry-2  10.3.2.102:8301  alive   server  0.8.1  2         →
 dc1
ostk003-uiserv-0           10.3.2.116:8301  alive   client  0.8.1  2         →
 dc1
ostk003-uiserv-1           10.3.2.117:8301  alive   client  0.8.1  2         →
 dc1
ostk003-vnflaf-services    10.3.1.249:8301  alive   client  0.8.1  2         →
 dc1
...
svc-2-httpd                10.3.2.35:8301   alive   client  0.8.1  2         →
 dc1
svc-2-sps                  10.3.2.111:8301  alive   client  0.8.1  2         →
 dc1
svc-2-sso                  10.3.2.113:8301  alive   client  0.8.1  2         →
 dc1
```

2. SSH to the VM you want.

   To access the VM, use the cloud-user user ID and include the path to the VM private key. The VM can be accessed using either the node identifier or it's IP address. For example:

```
[cloud-user@ostk003-emp-0 ~]$ ssh -i /var/tmp/vm_private_key cloud-user@10.3 →
.2.31
The authenticity of host 'haproxy (10.3.2.31)' can't be established.
RSA key fingerprint is b9:4f:ca:4f:bc:55:00:de:a8:77:e5:08:56:7c:db:98.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'haproxy,10.3.2.31' (RSA) to the list of known ho →
sts.
[cloud-user@haproxy ~]$
```

2.3 View Log Files and Dump Locations on a Virtual Machine

The following are details of log files available within each service in ENM.

Logs

All logs are configured to be forwarded to the Central Log Service. As such they are visible in Log Viewer using the ENM Launcher.

JBOSS Logs

All JBOSS logs are stored locally in `/ericsson/3pp/jboss/standalone/log`

3PP & System Logs

As standard, most 3PP and system logs are available locally in `/var/log`

Dumps

All application memory and core dump files are located in `/ericsson/enm/dumps`

3 Restarting a Service

3.1 Restart a Service on a Physical ENM Deployment

Prerequisites

— Root access to MS.

Steps

1. Establish the service instances installed on the ENM deployment using `grep` for a particular service instance:

```
[root@<MS> ~]# /opt/ericsson/enminst/bin/vcs.bsh --groups | grep <service_na →
me>
```

Example

```
[root@ieatlms4352 ~]#  /opt/ericsson/enminst/bin/vcs.bsh --groups | grep msp →
m

svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb2539-1 parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4373    parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4374    parallel vm ONLINE OK -
```

2. Restart the VCS service group:

```
/opt/ericsson/enminst/bin/vcs.bsh --restart -g <service_group> -s <system>
```

Note: The `-s` command restarts only one service at a time. To restart multiple services, repeat the command and modify the system name.

It is not recommended (unless specifically instructed) to restart more than one instance of a service at the same time. Restarting more than one instance of a service at the same time impacts the service availability and also results in some application specific consequences.

Example

```
/opt/ericsson/enminst/bin/vcs.bsh --restart -g Grp_CS_svc_cluster_mspm -s ie →
atrcxb4373

[root@ms-1 bin]# bash vcs.bsh --restart -g Grp_CS_svc_cluster_mspm -s ieatrc →
xb4373
2020-07-23 12:02:04.481 INFO hagrp_offline : Offlining 1 group(s)
2020-07-23 12:02:04.515 INFO hagrp_offline : Offlining Grp_CS_svc_cluster_ms →
pm on ieatrcxb4373
2020-07-23 12:02:04.807 INFO wait_vcs_state : Waiting for Grp_CS_svc_cluster →
_mspm to go OFFLINE on ieatrcxb4373 (timeout=1800)
2020-07-23 12:05:43.185 INFO wait_vcs_state : Group Grp_CS_svc_cluster_mspm  →
now OFFLINE on ieatrcxb4373 (3m:39s)
```

```
2020-07-23 12:05:43.817 INFO hagrp_online : Onlining 1 group(s)
2020-07-23 12:05:43.822 INFO online_services : Onlining Grp_CS_svc_cluster_m →
spm on ieatrcxb4373
2020-07-23 12:05:44.057 INFO wait_vcs_state : Waiting for Grp_CS_svc_cluster →
_mspm to go ONLINE on ieatrcxb4373 (timeout=4500)
2020-07-23 12:09:03.400 INFO wait_vcs_state : Group Grp_CS_svc_cluster_mspm  →
now ONLINE on ieatrcxb4373 (3m:19s)
[root@ms-1 bin]#
```

3. Verify if the service instance is `ONLINE`:

```
/opt/ericsson/enminst/bin/vcs.bsh --groups | grep mspm
```

Example

```
[root@ieatlms4352 ~]#  /opt/ericsson/enminst/bin/vcs.bsh --groups | grep msp →
m

svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb2539-1 parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4373    parallel vm ONLINE OK -
svc_cluster Grp_CS_svc_cluster_mspm ieatrcxb4374    parallel vm ONLINE OK -
```

4. After the service restarted in *Step 2* is `ONLINE`, you can repeat *Step 2* and *Step 3* to restart further instances of the service as per your requirement.

3.2 Restart a Service on an ENM on Cloud Deployment

Prerequisites

— User connected to EMP server.

Steps

1. Establish the service instances installed on the vENM deployment using grep for a particular service instance.

```
#consul members | grep <service name>
```

Example

```
#consul members | grep mscm
```

2. Connect to the VM of the service group by following *section 3.2* and trigger a healthcheck failure of the VM by killing consul.

```
#pkill consul
```

3. Verify if the service instance is ONLINE.

4. After the restarted service is ONLINE, repeat the preceding two steps to restart further instances of the service as per your requirement.

4 Configuring PIB Parameters

> To configure a Platform Integration Bridge (PIB) parameter, it is necessary to determine what environment you are working on and follow the task relevant to your environment.

4.1 Configuring PIB Parameters on a Physical ENM Deployment

To access the PIB (Platform Integration Bridge) script to update parameters for ENM applications, it is necessary to determine what environment you are working on and follow the task relevant to your environment. This task outlines the steps to read and configure PIB parameters on a physical ENM Deployment.

Prerequisites

— A command window is open and you have super user privileges.

— You are connected to the ENM MS as per the Connect to a Virtual Machine on a Physical ENM Deployment on page 2.

Steps

1. Find the hostname for the service instance:

   ```
   grep <service_name> /etc/hosts
   ```

2. Choose one of the returned hostnames for the next steps.

3. Navigate to the following directory:

   ```
   [root @ms-1 ~]# cd /ericsson/pib-scripts/etc/
   ```

4. Check a configuration parameter on sample VM:

   ```
   ./config.py read --app_server_address=<service VM hostname>:8080 --service_i →
   dentifier=<service_identifier_name> --name=<parameter_name>
   ```

   Note: `--service_identifier=<service_identifier_name>` is optional for this command.

   Example

   To check value of the `SMRS_ERBS_NoOf_BACKUP_FILES` parameter:

```
./config.py read --app_server_address=svc-1-smrsserv:8080 --name=SMRS_ERBS_N
oOf_BACKUP_FILES
```

5. Update a configuration parameter on a deployed VM:

```
./config.py update --app_server_address=<service VM hostname>:8080 --service
_identifier=<service_identifier_name> --name=<parameter_name> --value=<new_v
alue>
```

Note: `--service_identifier=<service_identifier_name>` is optional for this command.

Example

To update the `SMRS_ERBS_NoOf_BACKUP_FILES` value to 4:

```
./config.py update --app_server_address=svc-1-smrsserv:8080 --name=SMRS_ERBS
_NoOf_BACKUP_FILES --value=4
```

Results

You have updated an application parameter using the PIB script.

4.2 Configuring PIB Parameters on ENM on Cloud Deployment

To access the PIB (Platform Integration Bridge) script to update parameters for ENM applications, it is necessary to determine what environment you are working on and follow the task relevant to your environment. This task outlines the steps to read and configure PIB parameters on an ENM on Cloud Deployment.

Note: ENM concepts are explained in the *ENM Product Description*.

Prerequisites

— A command window is open and you have super user privileges.

— You are connected to an EMP VM using Connect to a Virtual Machine on an ENM on Cloud Deployment on page 3.

Steps

1. As cloud-user change to root:

```
[cloud-user@emp ~]$ sudo su -
[root@emp ~]#
```

2. Find the hostname for the service instance:

```
consul members|grep <service_name>
```

3. Choose one of the returned hostnames for the next steps.
4. Change directory to where the config.py script is located:

```
[root@emp ~]# cd /ericsson/pib-scripts/etc/
[root@gat-emp-0 etc]#
```

5. Read the current parameter value:

```
./config.py read --app_server_address=<service VM hostname>:8080 --service_i →
dentifier=<service_identifier_name> --name=<parameter_name>
```

6. Set the parameter to the required value:

```
./config.py update --app_server_address=<service VM hostname>:8080 --service →
_identifier=<service_identifier_name> --name=<parameter_name> --value=<new_v →
alue>
```

Results

You have updated an application parameter using the PIB script.

5 Configuration Management Administration Tasks

> Routine operation and maintenance tasks related to the administration of the Configuration Management applications.

5.1 Configure the Scheduled Cleanup of Export Jobs

> The Bulk Export service provides an automatic cleanup operation to delete Bulk export jobs and their associated export files.

The cleanup job runs at a configurable time and automatically deletes export jobs which are older than a configurable age. By default, the cleanup operation runs at 2am daily and deletes export jobs older than four days.

It is not possible to download Bulk export files when the associated export jobs have been deleted from the system. Therefore, if the export files need to be retained, download and save files to a well-known location before the scheduled cleanup deletes the export job.

Export job and associated export files are deleted from the system based on the applied settings.

The status messages associated with the deleted export jobs are removed from the command `cmedit export --status` result.

Note: For details on how to view and modify PIB parameters using the service name impexpserv, refer to section Configuring PIB Parameters.

To change settings, choose the parameter name and value based on the details in section Cleanup Bulk Export Jobs Parameters Table . To execute this operation, a basic understanding of Unix Cron expressions is necessary.

5.1.1 Cleanup Bulk Export Jobs Parameters Table

Table 1 Parameters to Change the Cleanup Settings

Parameter Name	Default Value	Warnings	Explanation	Value Range
`scheduledCleanupExportEnabled`	true	Setting this parameter to false requires manual cleanup of the Bulk CM export jobs.(1)	Enable or disable export cleanup scheduling.	true, false
`scheduledCleanupExportDayOfMonth`	*	If set to once a month, the share	Every day of the month.	* 1-31

Parameter Name	Default Value	Warnings	Explanation	Value Range
		can fill up with Bulk export files. It is recommended not to change the default value.	A specific numeric day of the month. A specific day of the month. Example: 2nd Fri. A negative number means the nth day or days. Before the end of the month. The last day of the month.	[1st, 2nd, 3rd, 4th, 5th, Last], [Sun, Mon, Tue, Wed, Thu, Fri,Sat] −7 to −1 Last
`scheduledCleanupExportDayOfWeek`	*	If set to once a week, the share can fill up with bulk export files. Allowing the process to run every day is highly recommended. It is recommended not to change the default value.	Every day of the week. Both `0` and 7 refer to Sunday.	* `0`-7 Sun, Mon, Tue, Wed, Thu, Fri, Sat
`scheduledCleanupExportYear`	*	It is recommended not to change the default value.	Every year. A particular calendar year by specifying a four-digit calendar year.	* 2015-9999
`scheduledCleanupExportMonth`	*		Every month. A particular numeric calendar month. A particular alphabetic calendar month.	* 1-12 Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
`scheduledCleanupExportHour`	2	Choose a time of day during which there is little or no load on the share. Default is 2am daily.	Every hour. A particular hour.	* `0`-23 (runs on the hour specified)
`scheduledCleanupExportMinute`	`0`		Every minute. A particular minute.	* `0`-59 (runs on the minute specified)
`scheduledCleanupExportSecond`	`0`		Every second. A particular second.	* `0`-59 (runs on the second specified)
`scheduledCleanupExportTime`	4 per time unit	The optimal value is determined based on the number of Bulk Exports that run per day and the compression type used to export.	Set the time limit to keep Bulk Export files.	1-2147483647
`scheduledCleanupExportUnit`	DAYS	Bulk Export files older than four days are deleted by default.	Set the time unit to keep Bulk Exports.	DAYS, HOURS, MINUTES

(1) For manual cleanup, refer to section *Reduce System Usage for Shared File System for Bulk Export in the ENM System Administrator Guide* page 298

Example 1

```
--name=scheduledCleanupExportEnabled --value="true"
```

5.2 Enable Pretty Format of Export Files

> Bulk Export service provides the capability of using a pretty format with line feeds and tabs (white spaces) for 3GPP XML export files.

If the export file parsing tool supports `minified` format, there is no need to enable this feature.

In export jobs with a huge number of files, the extra line feeds, and the white spaces, increase the disk space needed for an export job. The default setting is to disable the pretty format.

Note: For details on how to view and modify PIB parameters using the service name impexpserv, refer to section Configuring PIB Parameters.

Table 2 Parameters to Enable Pretty Format

Parameter Name	Default Value	Explanation	Value Range
`prettyFormatEnabled`	false	Enable or disable the pretty format.	true, false
`numberOfSpacesToIndent`	2	The number of white spaces to indent.	2.4. Values higher than 4 are not considered standard.

5.3 Configure the Scheduled Cleanup of Blocked Import and Export Jobs

> The Bulk import and export service provides a checking process to move blocked jobs to a *<Failed>* state . (Jobs stuck in *<Starting>* or *<Started>* state for a given time).

Note: For details on how to view and modify PIB parameters using the service name impexpserv, refer to section Configuring PIB Parameters.

To change settings, choose the parameter name and value based on the details in section Clean up Blocked Import and Export Jobs Parameters Tables. To execute this operation, a basic understanding of Unix Cron expressions is necessary.

5.3.1 Clean up Blocked Import and Export Jobs Parameters Tables

Table 3 Parameters to Control Time and Frequence to Clean up Blocked Import and Export Jobs

Parameter Name	Default Value	Warnings	Explanation	Value Range
scheduledUpdateBlockedJobsEnabled	true	Setting this to false disables the automatic update of blocked jobs.	Enable or disable import and export update of blocked jobs.	true, false
scheduledUpdateBlockedJobsDayOfMonth	*		Every day of the month. A specific numeric day of the month. A specific day of the month. For example, second Fri. A negative number means the nth day before the end of the month. Example: -1 means the last day of the month. The last day of the month.	* 1–31 [1st, 2nd, 3rd, 4th, 5th, Last],[Sun, Mon, Tue, Wed, Thu, Fri,Sat] –7 to –1 Last
scheduledUpdateBlockedJobsDayOfWeek	*		Every day of the week. 0–7 (both 0 and 7 refer to Sunday). Sun, Mon, Tue, Wed, Thu, Fri, Sat.	* 0–7 Sun, Mon, Tue, Wed, Thu, Fri, Sat
scheduledUpdateBlockedJobsYear	*	It is recommended not to change the default value.	Every year A particular calendar year by specifying a four-digit calendar year.	* 2015–9999
scheduledUpdateBlockedJobsMonth	*	It is recommended not to change the default value.	Every month. A particular numeric calendar month. A particular alphabetic calendar month.	* 1–12 Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.
scheduledUpdateBlockedJobsHour	2	Default is every 2 hours (*/2).	Every hour. A particular hour.	* 0–23 (runs on the hour specified)
scheduledUpdateBlockedJobsMinute	0		Every minute. A particular minute.	* 0–59 (runs on the minute specified)
scheduledUpdateBlockedJobsSecond	0		Every second. A particular second.	* 0–59 (runs on the second specified)
scheduledUpdateBlockedJobsTime	5 per time unit as below		Set the time limit to keep blocked import and export jobs as per the time unit below.	1–2147483647
scheduledUpdateBlockedJobsUnit	HOURS	Blocked import and export jobs older than 5 hours are moved to a FAILED state by default.	Set the time unit to keep blocked import and export jobs.	DAYS, MINUTES, HOURS

Example 2

```
--name=scheduledUpdateBlockedJobsDayOfMonth --value="*"
```

5.4 Update Schedule Parameters in the Export Service

> The Bulk Export service provides the capability to schedule bulk exports at a particular time. This section describes the parameters used to configure the scheduled exports, and how to update them.

Note: For details on how to view and modify PIB parameters using the service name impexpserv, refer to section Configuring PIB Parameters.

To change settings, choose the parameter name and value based on the details in section Bulk Export Time Schedule Parameters. To execute this operation, a basic understanding of Unix Cron expressions is necessary.

5.4.1 Bulk Export Time Schedule Parameters

Table 4 Parameters to Configure the Export Scheduler

Parameter Name	Description	Default Value	Allowable Values
`scheduledExportEnabled`	Enable or disable export scheduling.	false	true, false
`scheduledExportSecond`	One or more seconds within a minute.	`0`	`0`-59 (Unix Cron expressions are supported).
`scheduledExportMinute`	One or more minutes within an hour.	`0`	`0`–59 (Unix Cron expressions are supported).
`scheduledExportHour`	One or more hours within a day.	*/6	`0`–23 (Unix Cron expressions are supported).
`scheduledExportDayOfWeek`	One or more days within a week.	*	`0`–7 (Unix Cron expressions are supported) Mon, Tue, Wed, Thu, Fri, Sat (both `0` and 7 refer to Sunday).
`scheduledExportDayOfMonth`	One or more days within a month.	*	1–31 (Unix Cron expressions are supported). -7 to -1 (A negative number means the nth day before the last day of the month) Last. [1st, 2nd, 3rd, 4th, 5th, Last] [Sun, Mon, Tue, Wed, Thu, Fri, Sat]
`scheduledExportMonth`	One or more months within a year.	*	1–12 (Unix Cron expressions are supported) Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec
`scheduledExportYear`	A particular calendar year.	*	A four-digit calendar year.

Parameter Name	Description	Default Value	Allowable Values
scheduledExportNodes	Nodes to be exported	*	Node name, FDN, or a semi-colon separated list of nodes or FDNs. The user can specify the wildcard character (*) at the beginning and end of the node name. For example: — RBS* - Export all nodes which name starts with RBS. *RBS - Export all nodes which name ends with RBS. *RBS* - Export all nodes which name contains RBS. * - Export all nodes. The wildcard is not supported for FDNs.
scheduledExportType	Type of export	3GPP	3GPP, dynamic
scheduledExportConfiguration	Configuration	Live	Live
scheduledExportFilterName	Filter name		A predefined filter. Refer to ENM online help for export command, to generate a list of filters.
scheduledCompressionType	Compression type	ZIP	ZIP, GZIP, NONE
scheduledNeType	Network Element type of the Node Search Scope.		The value must match the neType attribute value in Network Element Managed Object.

Example 3

```
--name=scheduledExportEnabled --value=true
```

This enables export scheduling.

```
--name=scheduledExportHour --value="13"
```

```
--name=scheduledExportNodes --value="ApolloERBS1;ApolloERBS2"
```

5.5 Update Parameters in Configuration Management Service

For details on how to view and modify PIB parameters using service name cmserv, refer to section Configuring PIB Parameters.

Table 5 Parameters to Update the Configuration Management Service (Cmserv)

Attribute	Description	Default Value	Example of Allowable Values
maxBatchSizeCmWriter	Maximum batch size for the CM writer service. In ENM CLI, the write commands are done in batches. This figure indicates the number of nodes which are processed in each batch of a write command. Changing this number changes the number of MOs affected in each batch.	100	50
maxAffectedObjectsWithoutConfirmation	Max number of affected objects that does not require confirmation in set, action, delete operations for CM writer service.	10	10
maxAllowedNodesForWriterCommands	Max number of nodes on which it can execute set, action, delete operations for CM writer service. The value -1 disables the functionality.	-1	50
syncStatusInfoEnabled	A Boolean flag to enable/disable the output of the additional syncStatus attribute in CM CLI GET command output.	false	true, false
thresholdForBlackListedMoInContainmentQueries	Max number of NEs allowed in Containment queries for black listed MOs.	450	300
writerExtraLogging	Enable the extra logging on CM writer service to identify who made any changes on the node. **Note:** Enabling this function will increase logging frequency. Log limit maybe exceeded.	false	true, false

5.6 Read and Configure Parameters for Configuration Management Service

Read and configure parameters in the Configuration Management service (cmserv).

Note: For details on how to view and modify PIB parameters using service name cmserv, refer to section Configuring PIB Parameters.

To change settings, choose the parameter name and value based on the details in section Read and Configure Configuration Management Service Parameters

Table. To execute this operation, a basic understanding of Unix Cron expressions is necessary.

5.6.1 Read and Configure Configuration Management Service Parameters Table

Table 6 Parameters to Read and Configure Cmserv

Attribute	Description	Default Value	Command Example to Update the Value	Expected New Value
`Apg_cm_policy_flow_control_config`	Maximum number of contemporary full synchronization flows that can run on each mscmapg.	["FLOW_CONTROL_PERIOD:20","//APG_MED/ApgLargeNodeFlow/1.0.0:4"]	`config.py update --app_server_address=svc-1-mscmapg:8080 --name=apgPolicyFlowControlConfig --value=FLOW_CONTROL_PERIOD:20","//APG_MED/SyncApgLargeNodeFlow/1.0.0:4`	["FLOW_CONTROL_PERIOD:20","//APG_MED/ApgLargeNodeFlow/1.0.0:4"]
`com_ecim_policy_flow_control_config`	Maximum number of contemporary full synchronization flows that can run on each mscmce.	["FLOW_CONTROL_PERIOD:20","//MEDIATION/SyncSgsnNodeFlow/1.0.0:2","//MEDIATION/SyncSgsnNodeFlow/2.0.0:2","//MEDIATION/SyncRadioNodeFlow/1.0.0:10","//MEDIATION/SyncRadioNodeFlow/2.0.0:10","//MEDIATION/SyncRadioNodeFlow/3.0.0:10","//MEDIATION/SyncRadioNodeFlow/4.0.0:10","//COM_ECIM_MED/SyncComLargeNodeFlow/1.0.0:1","//IPOSOI_MED/IposOiSyncNodeFlow/1.0.0:1","//IPOSOI_MED/IposOiSyncNodeFlow/2.0.0:1","//CBPOI_MED/SmallYangSyncNodeFlow/1.0.0:10","//CBPOI_MED/SmallYangSyncNodeFlow/1.1.0:10","//IPOSOI_MED/SmallYangIposOiSyncNodeFlow/1.0.0:10"]	`config.py update --app_server_address=svc-1-mscmce:8080 --name=com_ecim_policy_flow_control_config --value=FLOW_CONTROL_PERIOD:20","//MEDIATION/SyncSgsnNodeFlow/1.0.0:2","//MEDIATION/SyncSgsnNodeFlow/2.0.0:2","//MEDIATION/SyncRadioNodeFlow/1.0.0:10","//MEDIATION/SyncRadioNodeFlow/2.0.0:10","//MEDIATION/SyncRadioNodeFlow/3.0.0:10","//MEDIATION/SyncRadioNodeFlow/4.0.0:10","//COM_ECIM_MED/SyncComLargeNodeFlow/1.0.0:1","//IPOSOI_MED/IposOiSyncNodeFlow/1.0.0:1","//IPOSOI_MED/IposOiSyncNodeFlow/2.0.0:1","//CBPOI_MED/SmallYangSyncNodeFlow/1.0.0:10,"//CBPOI_MED/SmallYangSyncNodeFlow/1.1.0:10,"//IPOSOI_MED/SmallYangIposOiSyncNodeFlow/1.0.0:10` **Note:** It is not possible to change only one of the parameters. To do that, write new parameter	["FLOW_CONTROL_PERIOD:20","//MEDIATION/SyncSgsnNodeFlow/1.0.0:2","//MEDIATION/SyncSgsnNodeFlow/2.0.0:2","//MEDIATION/SyncRadioNodeFlow/1.0.0:10","//MEDIATION/SyncRadioNodeFlow/2.0.0:10","//MEDIATION/SyncRadioNodeFlow/3.0.0:10","//MEDIATION/SyncRadioNodeFlow/4.0.0:10","//COM_ECIM_MED/SyncComLargeNodeFlow/1.0.0:1","//IPOSOI_MED/IposOiSyncNodeFlow/1.0.0:1","//IPOSOI_MED/IposOiSyncNodeFlow/2.0.0:1","//CBPOI_MED/SmallYangSyncNodeFlow/1.0.0:10","//CBPOI_MED/SmallYangSyncNodeFlow/1.1.0:10","//IPOSOI_MED/SmallYangIposOiSyncNodeFlow/1.0.0:10"]

Attribute	Description	Default Value	Command Example to Update the Value	Expected New Value
			value with all the other parameter values as they currently are.	
dpsTimesRetryOnError	Some attempts to store data into DPS, for example during synchronization process.	3 (Allowed range 1–10)	config.py update --name=dpsTimesRetryOnError --value=5 --app_server_address=svc-1-mscmce:8080	5
dpsUploadBatchSize	Batch size for saving number of Managed Objects in DPS, while sync use case.	1000 (Allowed range 2–2000)	config.py update --name=dpsUploadBatchSize --value=500 --app_server_address=svc-1-mscmce:8080	500
netconfSocketConnectionTimeout	Default socket time-out (in milliseconds) for socket connection time-out.	3000	config.py update --name=netconfSocketConnectionTimeout --value=10000 --app_server_address=svc-1-mscmce:8080	10000
netconfSocketTimeout	Default socket time-out (in milliseconds) for a netconf manager.	10000	config.py update --name=netconfSocketTimeout --value=100000 --app_server_address=svc-1-mscmce:8080	100000
timeoutdpsUpload	Time-out (in milliseconds) for saving batch in DPS, while sync use case.	120000 (Allowed range 60000–300000)	config.py update --name=timeoutdpsUpload --value=300000 --app_server_address=svc-1-mscmce:8080	300000

5.7 Disable Bulk CM Export of Non-Synchronized Nodes

By default, CM Bulk Export exports all specified nodes regardless of synchronization status, therefore non-synchronized nodes are exported. These include nodes that were never synchronized, nodes that are currently synchronizing, and nodes that are currently non-synchronized. It is possible to configure the default behavior so that only synchronized nodes are exported. (That is nodes with a CmFunction with syncStatus attribute equal to SYNCHRONIZED)

Note: For details on how to view and modify PIB parameters using service name impexpserv, refer to section Configuring PIB Parameters.

Table 7 Parameters to Disable Bulk CM Export of Non-Synchronized Nodes

Parameter Name	Default Value	Warnings	Explanation	Value Range
exportNonSynchronizedNodes	true		Enable (true) or disable (false) the export of non-synchronized nodes.	true, false

5.8 Enable Bulk CM Export to Ignore Non-persistent Attributes for Non-Synchronized Nodes

By default, CM Bulk Export fails to export non-synchronized nodes for jobs with a user-defined filter containing any non-persistent attribute. (That is for nodes with a CmFunction with `syncStatus` attribute not equal to SYNCHRONIZED).

It is possible to change this default behavior so that only the persistent attributes will be exported for non-synchronized nodes (that is, ignoring the non-persistent attributes in the user-defined filter for non-synchronized nodes).

Note: For details on how to view and modify PIB parameters, using service name `impexpserv`, refer to Configuring PIB Parameters on page 8.

Table 8 Parameter to Enable Bulk CM Export to ignore Non-persistent attributes of Non-Synchronized Nodes

Parameter Name	Default Value	Warnings	Explanation	Value Range
bulkCmExport_ignoreNPAforNonSynchedNodes	false		Enable (true) or disable (false) export of only persistent attributes for non-synchronized Nodes	false , true

5.9 Enable Bulk CM Export of CPP Inventory MOs

For nodes based on the CPP platform, the Inventory MOs are modeled directly under the MeContext MO. It is possible to configure if the unfiltered Bulk CM export includes these MOs. The default is not to include them.

Note: For details on how to view and modify PIB parameters, refer to section Configuring PIB Parameters.

Table 9 Parameters to Enable Bulk CM Export of CPP Inventory MOs

Parameter Name	Default Value	Warnings	Explanation	Value Range
exportCppInventoryMos	false		Enable (true) or disable (false) the export of Inventory	true, false

Parameter Name	Default Value	Warnings	Explanation	Value Range
			MOs for node based on CPP platform.	

5.10 Configure the Default Behavior of Enum Translation for CM Export

The Enum translation behavior for CM Bulk Export determines how Enum values are represented in generated export files. (That is, either as String literals (default) or as Integers). It is possible to configure the default behavior so that Enum values are exported as Integers. The user is able to override this configured default behavior for each export job. Do this by specifying a parameter in the ENM CLI or in the Bulk REST API.

Note: For details on how to view and modify PIB parameters using service name impexpserv, refer to section Configuring PIB Parameters.

Parameters to Configure the Default Behavior of Enum Translation for CM Export

Table 10 Parameters to Enable Bulk CM Export of CPP Inventory MOs

Parameter Name	Default Value	Warnings	Explanation	Value Range
`enumTranslate`	true		If value is set to 'true', Enum values are exported as String literals. If value is set to 'false', Enum values are exported as Integers.	true, false

5.11 Configure the Default Data Categories of the 3GPP CM Export

By default, when the user does not specify the data categories for an export, the 3GPP CM Export will only export MOs from the `NETWORK_RESOURCE_DATA` data category.

To change this default behavior configure the parameter outlined in Table 11. By setting the parameter to `true` the 3GPP CM Export for 3GPP includes `ENM_DATA` and `NETWORK_RESOURCE_DATA` data categories.

Note: For details on how to view and modify PIB parameters using service name `impexpserv`, refer to Configuring PIB Parameters on page 8.

Table 11 Parameters to Enable the inclusion by default of ENM_DATA in the 3GPP CM Export.

Parameter Name	Default Value	Warnings	Description	Value Range
bulkCmExport_includeEnmData3gpp	false		If value is set to 'true', ENM_DATA and NETWORK_RESOURCE_DATA will be exported. If value is set to 'false', NETWORK_RESOURCE_DATA only will be exported.	false, true

5.12 Configure the Default MO Order of CM Export

By default, the CM Bulk Export is not guaranteeing the MO instances FDN order.

It is possible to configure this default ordering behaviour to an alphabetical sort order of the MO instances FDNs, by setting the parameter outlined in Table 12 to true. This may have an impact on the performance of the export job.

Note: For details on how to view and modify PIB parameters using service name `impexpserv`, see Configuring PIB Parameters on page 8.

Table 12 Parameter to Enable the alphabetical order of the MOs Instances FDN

Parameter Name	Default Value	Warnings	Description	Value Range
bulkCmExport_orderMOsByFDN	false		If value is set to 'true', MO instances FDNs are sorted in an ascending alphabetical order	false, true

5.13 Configure Bulk Import Service

This section provides the parameters to configure the Bulk Import service.

Note: The parameters listed in the following table below are applicable to both physical and cloud environments. The steps and examples show how to configure the parameters on a physical environment.

For details on how to view and modify PIB parameters using service name impexpserv, refer to Configuring PIB Parameters on ENM on Cloud Deployment.

Table 13 Configuration Parameters in CM Bulk Import

Parameter Name	Description	Default Value	Value Range
bulkCmImport_importFileSizeLimit	Set import file size limit in megabytes (MB).	100	1–512
bulkCmImport_numberOfOperationsPerTransaction	Set the number of import operations per node allowed in a single transaction for non-BSC nodes.	300	1–1000
bulkCmImport_numberOfBscNodeOperationsPerTransaction	Set the number of import operations per node allowed in a single transaction for a BSC Node.	2048	1–2048
bulkCmImport_numberOfOperationRetries	Set the number of retry attempts for failed import operations.	2	0–4
bulkCmImport_importJobDataRetentionPeriod	The number of days to retain import jobs. Import jobs older than the specified number of days are deleted along with their associated files excluding the scheduled jobs. **Note:** When decreasing the value, all jobs that fall outside the new retention period are automatically deleted excluding the scheduled jobs.	90	1–90

Parameter Name	Description	Default Value	Value Range
`bulkCmImport_deleteFileAfterParsing`	When the Import file has been parsed, it can be deleted from the ENM file system. However, the system administrator (or operator) can decide whether to remove or retain the import file after parsing. File will be deleted only when an Import Job is created through an asynchronous interface, and parameter is set to true. **Note:** File deletion is attempted on best effort basis. Failure during file deletion would not impact import job progress.	false	false, true

5.14 Enabling Unsecure FTP

This script is executed from each SMRS VM.

Prerequisites

— ENM has been deployed.

— User has root access to SMRS VM

Steps

1. Log on to the ENM Management Server as the litp-admin user and switch to the root user:

If password authentication is disabled for the litp-admin user, then refer to *Log On to the MS when Password Authentication is Disabled* in the *LITP Node Hardening Instructions*.

```
ssh litp-admin@<ENM Management Server IP Address>
su -
```

Example:

```
# ssh litp-admin@10.10.10.10
############ WARNING #############

This system is for authorised use only. By using this system you consent to
monitoring and data collection.

#################################
litp-admin@10.10.10.10's password:
[litp-admin@ms-1 ~]$ su -
[root@ms-1~]#
```

2. Logon to Each SMRS VM and switch to root user to execute script

Example:

```
[root@ms-1~]# ssh -i /root/.ssh/vm_private_key cloud-user@svc-1-smrsserv

Last login: Mon Dec 24 12:47:00 2018 from ms-1

[cloud-user@svc-1-smrsserv ~]$ sudo su

[root@svc-1-smrsserv cloud-user]# cd /opt/ericsson/ERICsmrsservice_CXP903075
5/

[root@svc-1-smrsserv ERICsmrsservice_CXP9030755]# ll

total 16

-r-xr-x---. 1 root root  895 Dec 24 08:44 disable_unsecured_ftp.sh

-r-xr-x---. 1 root root 3214 Dec 20 10:01 enable_unsecured_ftp.sh

[root@svc-1-smrsserv ERICsmrsservice_CXP9030755]#
```

3. Enter the command to execute the enable_unsecured_ftp script to enable unsecure ftp:

```
/opt/ericsson/ERICsmrsservice_CXP9030755/enable_unsecured_ftp.sh
```

Example:

```
[root@svc-1-smrsserv cloud-user]# /opt/ericsson/ERICsmrsservice_CXP9030755/e
nable_unsecured_ftp.sh
UNSECURED_FTP_ENABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP9030755
/enable_unsecured_ftp.sh ): Set SE Linux policy to allow access to FTP HOME
DIR
UNSECURED_FTP_ENABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP9030755
/enable_unsecured_ftp.sh ): Set SE Linux policy to allow ftpd use nfs
UNSECURED_FTP_ENABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP9030755
/enable_unsecured_ftp.sh ): Configuring chroot for MINI-LINK-Indoor ftp User
s
UNSECURED_FTP_ENABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP9030755
/enable_unsecured_ftp.sh ): Remove default configuration files ...
UNSECURED_FTP_ENABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP9030755
/enable_unsecured_ftp.sh ): Creating configuration files for unsecured ftp .
```

```
..
Shutting down vsftpd: [ OK ]
Starting vsftpd for ftp: [ OK ]
Starting vsftpd for ftpes: [ OK ]
Starting vsftpd for ftpes_ipv6: [ OK ]
Starting vsftpd for ftp_ipv6: [ OK ]
UNSECURED_FTP_ENABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP9030755 →
/enable_unsecured_ftp.sh ): Creating backup of unsecured ftp conf files
[root@svc-1-smrsserv cloud-user]#
```

4. Enter the command to execute the disable_unsecured_ftp script to disable unsecure ftp:

```
/opt/ericsson/ERICsmrsservice_CXP9030755/disable_unsecured_ftp.sh
```

Example:

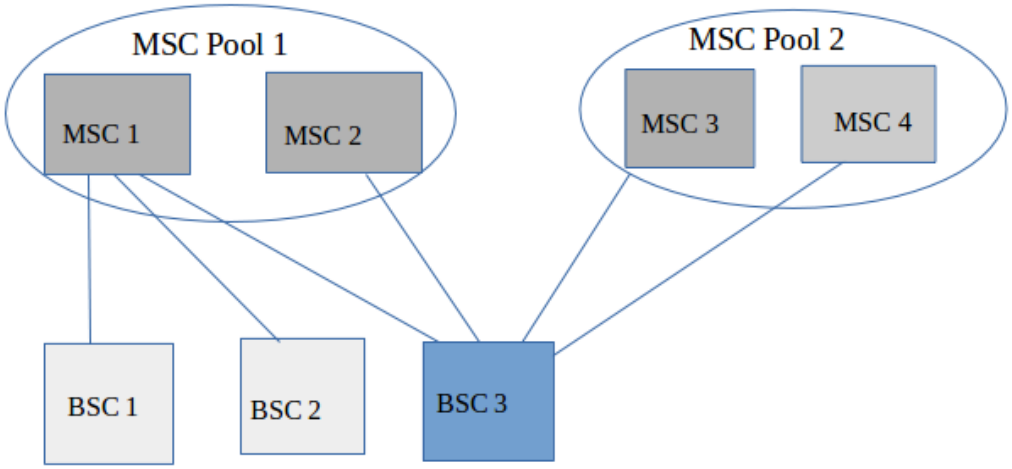
```
[root@svc-1-smrsserv cloud-user]# /opt/ericsson/ERICsmrsservice_CXP9030755/d →
isable_unsecured_ftp.sh

UNSECURED_FTP_DISABLE: INFORMATION ( /opt/ericsson/ERICsmrsservice_CXP903075 →
5/disable_unsecured_ftp.sh ): Removing unsecure ftp conf files....
Shutting down vsftpd: [ OK ]
Starting vsftpd for ftpes: [ OK ]
Starting vsftpd for ftpes_ipv6: [ OK ]
[root@svc-1-smrsserv cloud-user]#
```

5.15 MSC Pool

An MSC Pool is a group of MSCs (MSC Pool Members), sharing in parallel, the traffic generated from one and only one MSC Pool Area. All MSCs in the same pool are connected to all the RAN (RNCs/BSCs) nodes in the related MSC Pool Area.

5.15.1 MSC Pool Area

The MSC pool area is a collection of BSC or RNC service areas, or both, that are served by one or more MSC nodes in parallel that share the traffic from the pool area. An MSC pool area is an area within which an MS can roam without a need to change the serving MSC node. A BSC/RNC service area must belong completely to the same one or more MSC pool area. All the MSC nodes in such a pool area share the responsibility of handling all the MSCs, in all the LAs of the pool area. Each MS can be served by any MSC within a pool area. Once the subscriber has entered a new pool area, in normal operation, the subscriber shall be served by the same MSC as long as the MS roams within the pool area.



5.15.2 MSC Pool Management

5.15.2.1 Create an MSC Pool

Note: MSC Pool names have to be unique across all networks of an operator, for example when moving MSC Pools management from one ENM System to another, or in a N:1 Geographical Redundant ENM setup.

Command

To create an MSC pool use the following command:

```
cmedit create Pool=<pool name> PoolId=<pool name>,PoolType=MSC_POOL -namespace=O
SS_NE_DEF -version=2.0.0
```

Example

```
»cmedit create Pool='Pool2' PoolId="Pool2",PoolType=MSC_POOL -namespace=OSS_NE_DEF -version=2.0.0
 FDN : Pool=Pool2
 PoolId : Pool2
 poolType : MSC_POOL


 1 instance(s) updated
```

5.15.2.2 View an MSC Pool

Command

To view an MSC pool use the following command::

```
cmedit get * Pool
```

Example

```
»cmedit get * Pool
 FDN : Pool=Pool1

 FDN : Pool=Pool2

 FDN : Pool=Pool3

 FDN : Pool=Pool5


 4 instance(s)
```

5.15.2.3 Delete an MSC Pool

Command

To delete an MSC pool use the following command:

```
cmedit delete Pool=<pool name> -ALL --force
```

Example

```
»cmedit create Pool="Pool6" PoolId="Pool6",PoolType=MSC_POOL -namespace=OSS_NE_DEF -version=2.0.0
 FDN : Pool=Pool6
 PoolId : Pool6
 poolType : MSC_POOL


 1 instance(s) updated
»cmedit delete Pool=Pool6 -ALL --force
 SUCCESS FDN : Pool=Pool6


 1 instance(s) deleted
»cmedit get * Pool
 FDN : Pool=Pool5


 1 instance(s)
»cmedit delete Pool=Pool5 -ALL --force
 SUCCESS FDN : Pool=Pool5


 1 instance(s) deleted
»cmedit get * Pool

 0 instance(s)
```

Note:

- MSC in POOL feature is supported for MSC neTypes MSC-BC-BSP, MSC-DB-BSP, MSC-BC-IS and vMSC only.
- To delete an MSC Pool, all associations to MSC and BSC network elements must be removed first, then proceed to MSC Pool deletion.
- To update an MSC Pool name, first remove that MSC Pool, then create a new MSC Pool with the new name.

5.15.2.4 Associate an MSC to an MSC Pool

Command

To associate an MSC with an MSC pool use the following command:

```
cmedit set NetworkElement=<node name> poolRefs = [Pool=<pool name>]
```

Example

```
1 instance(s)
»cmedit set NetworkElement=MSC07 poolRefs = [Pool=Pool2]
SUCCESS FDN : NetworkElement=MSC07


1 instance(s) updated
»cmedit get NetworkElement=MSC07
FDN : NetworkElement=MSC07
failedSoftwareSyncsCount : 0
lastSuccessfulSoftwareSync : null
managedBy : null
managementState : NORMAL
neProductVersion : null
networkElementId : MSC07
neType : MSC-BC-BSP
nodeModelIdentity : null
ossModelIdentity :
ossPrefix : SubNetwork=NETSimW,MeContext=MSC07
platformType : ECIM
poolRefs : [Pool=Pool2]
release : null
softwareSyncStatus : UNINITIALIZED
technologyDomain : null
timeZone : IST
userLabel :
utcOffset : null


1 instance(s)
```

5.15.2.5 View Associated MSC Pools of an MSC

Command

To view MSC pools associated with an MSC use the following command:

```
cmedit get NetworkElement=<node name>
```

Example

```
»cmedit get NetworkElement=MSC07
 FDN : NetworkElement=MSC07
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 neProductVersion : null
 networkElementId : MSC07
 neType : MSC-BC-BSP
 nodeModelIdentity : null
 ossModelIdentity :
 ossPrefix : MeContext=MSC07
 platformType : ECIM
 poolRefs : [Pool=Pool1]
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
 userLabel :
 utcOffset : null


 1 instance(s)
```

5.15.2.6 Associate a BSC to Multiple MSC Pools

Command

To associate a BSC with multiple MSC pools use the following command:

```
cmedit set NetworkElement=<node name> poolRefs =[Pool=<pool name>,Pool=<another pool name>]
```

Example

```
»cmedit set NetworkElement=MSC07 poolRefs =[Pool=Pool3,Pool=Pool4]
 SUCCESS FDN : NetworkElement=MSC07


 1 instance(s) updated
»cmedit get NetworkElement=MSC07
 FDN : NetworkElement=MSC07
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 neProductVersion : null
 networkElementId : MSC07
 neType : MSC-BC-BSP
 nodeModelIdentity : null
 ossModelIdentity :
 ossPrefix : MeContext=MSC07
 platformType : ECIM
 poolRefs : [Pool=Pool3, Pool=Pool4]
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
 userLabel :
 utcOffset : null
```

5.15.2.7 Delete MSC Pool Association from BSC with Multiple Associations

Command

To delete an MSC pool association from a BSC that has multiple MSC pools associated use the following command:

```
cmedit set NetworkElement=<node name> mscPoolRefs=[Pool=<pool name>]
```

Example

In this example, 'Pool3' association is being removed.

```
»cmedit set NetworkElement=GSM01BSC01 mscPoolRefs=[Pool=Pool3]
 SUCCESS FDN : NetworkElement=GSM01BSC01


 1 instance(s) updated
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : null
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : [Pool=Pool3]
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
 userLabel :
```

5.15.2.8 Delete All Associations of a BSC

Command

To delete all MSC pool associations of a BSC use the following command:

```
cmedit set NetworkElement=<node name> mscPoolRefs=[]
```

Example

image

```
»cmedit set NetworkElement=GSM01BSC01 mscPoolRefs=[]
 SUCCESS FDN : NetworkElement=GSM01BSC01


 1 instance(s) updated
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : null
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : []
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
```

5.15.2.9 Update Different MSCs Connected to a BSC

Command

To update different MSCs connected to a BSC use the following command:

```
cmedit set NetworkElement=<BSC node name> ConnectedMscs=[NetworkElement=<MSC nod →
e name>]
```

Example

```
»cmedit set NetworkElement=GSM01BSC01 connectedMscs=[NetworkElement=MSC07]
 SUCCESS FDN : NetworkElement=GSM01BSC01


 1 instance(s) updated
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : [NetworkElement=MSC07]
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : []
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
```

5.15.2.10 View Different MSCs Connected to a BSC

Command

To view different MSCs connected to a BSC use the following command:

```
cmedit get NetworkElement=<BSC node name>
```

Example

```
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : [NetworkElement=MSC07]
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : []
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
 userLabel :
 utcOffset : null


 1 instance(s)
```

5.15.2.11 Update Multiple MSCs Connected to a BSC

Command

To update multiple MSCs connected to a BSC use the following command:

```
cmedit set NetworkElement=<BSC node name>  connectedMscs=[NetworkElement=<MSC no →
de name>,NetworkElement=<another MSC node name> ]
```

Example

```
»cmedit set NetworkElement=GSM01BSC01 connectedMscs=[NetworkElement=MSC07,NetworkElement=MSC08]
 SUCCESS FDN : NetworkElement=GSM01BSC01


 1 instance(s) updated
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : [NetworkElement=MSC07, NetworkElement=MSC08]
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : []
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
 userLabel :
```

5.15.2.12 Remove Specific MSCs Connected to a BSC

Command

To remove specific MSCs connected to a BSC use the following command:

```
cmedit set NetworkElement=<BSC node name> connectedMscs=[NetworkElement=<MSC nod →
e name>]
```

Example

In this example, network element 'MSC07' is removed and 'MSC08' remains connected.

```
»cmedit set NetworkElement=GSM01BSC01 connectedMscs=[NetworkElement=MSC08]
 SUCCESS FDN : NetworkElement=GSM01BSC01


 1 instance(s) updated
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : [NetworkElement=MSC08]
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : []
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
```

5.15.2.13 Delete All MSCs Connected to a BSC

Command

To delete all MSCs connected to a BSC use the following command:

```
cmedit set NetworkElement=<BSC node name> connectedMscs=[]
```

Example

```
»cmedit set NetworkElement=GSM01BSC01 connectedMscs=[]
 SUCCESS FDN : NetworkElement=GSM01BSC01


 1 instance(s) updated
»cmedit get NetworkElement=GSM01BSC01
 FDN : NetworkElement=GSM01BSC01
 connectedMsc : null
 connectedMscs : []
 failedSoftwareSyncsCount : 0
 lastSuccessfulSoftwareSync : null
 managedBy : null
 managementState : NORMAL
 mscPoolRefs : []
 neProductVersion : null
 networkElementId : GSM01BSC01
 neType : BSC
 nodeModelIdentity : null
 ossModelIdentity : BSC-G17.Q4-R1C-APG43L-3.4.0-R5A
 ossPrefix : MeContext=GSM01BSC01
 platformType : ECIM
 release : null
 softwareSyncStatus : UNINITIALIZED
 technologyDomain : null
 timeZone : IST
 userLabel :
```

For a BSC, the associated MSC Pool names can be looked up, using the attribute `mscPoolRefs`. Then, find all MSCs which belong to the respective MSC pool. To find out the members of an MSC Pool, get the pool associations of the respective network element.

Note: The `connectedMscs` attribute can hold MSCs which are part of a MSC Pool or not. ENM treats them as non-pooled MSCs, that is not part of any MSC Pool.

5.16 Managing SNMP Connection Parameters for MINI-LINK Outdoor Nodes

5.16.1 Changing SNMP Configuration from SNMPv2C to SNMPv3

Prerequisites

— Node must be added and configured with SNMPv2c.

Steps

1. If node is synchronized, unsync the node by disabling CM supervision.

```
cmedit set NetworkElement=<nodeName>,CmNodeHeartbeatSupervision=1 active=fal →
se
```

2. Set SNMP connectivity to SNMPv3.

   Example:

```
cmedit set NetworkElement=<nodeName>,MINILINKOutdoorConnectivityInformation= →
1 snmpVersion="SNMP_V3"
```

3. Update SNMP security credentials if not already updated as part of add node.

```
secadm snmp authpriv --auth_algo <Auth algoritham> --auth_password <authpass →
word> --priv_algo <Priv algoritham> --priv_password <privpassword> -n <nodeN →
ame>
```

4. Makesure that snmpv3 already configured on outdoor node. Otherwise, connectivity is lost and to remove snmpv2c connectivity on node, refer to SNMP User Guide MINI-LINK 6352.

5. Enable CM supervision.

```
cmedit set NetworkElement=<nodeName>,CmNodeHeartbeatSupervision=1 active=tru →
e
```

6. Verify CM sync is successful.

```
cmedit get NetworkElement=<nodeName>,CmFunction=1
```

5.16.2 Changing SNMP configuraion from SNMPv3 to SNMPv2C

Prerequisites

— Node must be added and configured with SNMPv3.

Steps

1. If node is synchronized, unsync the node by disabling CM supervision.

```
cmedit set NetworkElement=<nodeName>,CmNodeHeartbeatSupervision=1 active=fal →
se
```

2. Set SNMP connectivity to SNMPv2c.

   Example:

```
cmedit set NetworkElement=<nodeName>,MINILINKOutdoorConnectivityInformation=
1 snmpVersion="SNMP_V2C"
```

3. Update SNMP security credentials if not already updated as part of add node.

```
secadm snmp authpriv --auth_algo <Auth algoritham> --auth_password <authpass
word> --priv_algo <Priv algoritham> --priv_password <privpassword> -n <nodeN
ame>
```

4. Make sure that SNMPv2c is already configured on the outdoor node. Otherwise, connectivity is lost and to remove snmpv3 connectivity on a node, refer to SNMP User Guide MINI-LINK 6352.

5. Enable CM supervision.

```
cmedit set NetworkElement=<nodeName>,CmNodeHeartbeatSupervision=1 active=tru
e
```

6. Verify CM sync is successful.

```
cmedit get NetworkElement=<nodeName>,CmFunction=1
```

5.17 Configuring PIB Parameters to set a Map Provider

There are three PIB parameters that control the map provider used by Network Viewer and Add Node:

Parameter Name	Type	Default	Value Description
`enabledMapProvider`	String	None	Name of the chosen provider of geographical map. For example, OpenStreetMap.
`mapProviderUrl`	String	<null>	A generic address able to identify in parametric way all the tiles of a certain geographical extension of a map.
`mapProviderAccessToken`	String	<empty string>	An access token provides access to map provider resources on behalf of a user.

Note: If the used map provider is free, the string `mapProviderAccessToken` must be `empty` as per default value. If the map provider is for payment, the string must be filled with the related token value.

Maps must be compatible with Leaflet library, version 1.0.4.

5.17.1 PIB Parameter Handling

Note: For details on how to view and modify PIB parameters using service name netex, see Configuring PIB Parameters on page 8.

5.18 Configure CM File Upload Timeout and Retry Parameters for SIU02/TCU02 Nodes

As part of SIU02/TCU02 node CM synchronization, by default CM mediation retries by polling node for 5 seconds to complete CM file upload to ENM and report the transfer status before aborting the operation on the node.

In some scenarios, due to node or network conditions, if the default 5 seconds is not enough for completing the file transfer, then the following PIB Parameters can be changed to increase the timeout.

Note: For details on how to view and modify PIB parameters, refer to section Configuring PIB Parameters on page 8.

Table 14

Parameter Name	Default Value	Warning	Explanation	Value Range
`stnCmUploadWaitIntervalInSeconds`	2	—	Retry wait interval in seconds used by CM STN mediation to verify the upload status on the node.	Min: 2 Max: 4
`stnCmUploadRetryCount`	5	—	Retry count for retrieving upload status on node by CM STN mediation.	Min: 4 Max: 5

6 Software and Hardware Management Administration Tasks

This section contains the routine operation and maintenance tasks related to the administration of Software and Hardware Management applications.

6.1 Configurable Parameters for Jobs Housekeeping

SHM enables you to create jobs to administer the nodes in your network. SHM provides housekeeping policies to ensure that, over time, the completed jobs do not cause data storage space issues.

Housekeeping Policies

The housekeeping of the jobs runs on a daily basis at 5 AM.

Schedule Time Parameters

Jobs Housekeeping policy contains job age criteria and job count criteria.

First, jobs cleanup is done based on age criteria. If the count of jobs is still more than specified maximum count per job type, then the clean is done until job count reaches specified maximum count per job type.

1. Job age criteria

   When the number of jobs exceeds the specified number of days, older jobs are deleted. The retention policy parameters used for the housekeeping jobs and the maximum age of a job vary from each job type.

2. Job Count criteria

   When the number of jobs exceeds the specified count, older jobs are deleted. The retention policy parameters used for the Housekeeping jobs and the job count vary from each job type.

Daily Schedule Time of Jobs housekeeping can be determined from the following two configurable parameters:

— `DAILY_SCHEDULE_TIME_IN_HOURS_FOR_JOB_HOUSEKEEPING` - It determines the hours of scheduled time of housekeeping.

— `DAILY_SCHEDULE_TIME_IN_MINUTES_FOR_JOB_HOUSEKEEPING` - It determines the minutes of scheduled time of housekeeping.

For example, if the value of the first parameter is 10 and the value of the second parameter is 30, the daily schedule time of Jobs House Keeping is 10h 30min.

6.2 Configuration Parameters for Housekeeping Modeled in SHM

All the parameters listed in Table 15 are configurable.

Note: For details on how to view and modify PIB parameters using service name shmcoreserv, see *Configuring PIB Parameters*.

Table 15 Configuration Parameters for Housekeeping Modeled in SHM

Parameter	Type	Scope	Default Value	Description
DAILY_SCHEDULE_TIME_IN_HOURS_FOR_JOB_HOUSEKEEPING	int	SERVICE	5	Hours of Daily Schedule Time when housekeeping is kicked off. The allowed values for this parameter are: [Min value = 0 hours][Max value = 23 hours]
DAILY_SCHEDULE_TIME_IN_MINUTES_FOR_JOB_HOUSEKEEPING	int	SERVICE	0	Minutes of Daily Schedule Time when housekeeping is kicked off. The allowed values for this parameter are: [Max value = 59 min] [Min value = 0 min]
BACKUP_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of backup jobs exceeds the specified number of days, older jobs are deleted.
UPGRADE_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	180 days	If the number of upgrade jobs exceeds the specified number of days, older jobs are deleted.
LICENSE_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of license jobs exceeds the specified number of days, older jobs are deleted.
RESTORE_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of restore jobs exceeds the specified number of days, older jobs are deleted.
DELETE_BACKUP_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of delete backup jobs exceeds the specified number of days, older jobs are deleted.
BACKUP_HOUSEKEEPING_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of backup housekeeping jobs exceeds the specified number of days, older jobs are deleted.

Parameter	Type	Scope	Default Value	Description
DELETE_UPGRADEPACKAGE_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of delete upgradepackage jobs exceeds the specified number of days, older jobs are deleted.
NODERESTART_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of NODERESTART jobs exceeds the specified number of days, older jobs are deleted.
DEFAULT_JOB_AGE_FOR_JOB_HOUSEKEEPING	int	SERVICE	30 days	If the number of other SHM jobs without configured age criteria exceeds the specified number of days, older jobs are deleted.
DELETE_UPGRADEPACKAGE_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	60	Maximum count of delete upgradepackage jobs. If the number of delete upgradepackage jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
BACKUP_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	200	Maximum count of backup jobs. If the number of backup jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
UPGRADE_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	200	Maximum count of upgrade jobs. If the number of upgrade jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
LICENSE_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	60	Maximum count of license jobs. If the number of license jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
RESTORE_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	60	Maximum count of restore jobs. If the number of restore jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
DELETE_BACKUP_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	60	Maximum count of delete backup jobs. If the number of delete backup jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
BACKUP_HOUSEKEEPING_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	60	Maximum count of backup housekeeping jobs. If the number of backup housekeeping jobs exceeds the specified count, older jobs are deleted as part of housekeeping.

Parameter	Type	Scope	Default Value	Description
NODERESTART_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	60	Maximum count of NODERESTART jobs. If the number of NODERESTART jobs exceeds the specified count, older jobs are deleted as part of housekeeping.
DEFAULT_JOB_COUNT_FOR_JOB_HOUSEKEEPING	int	SERVICE	30	Maximum count of other SHM jobs without configured housekeeping criteria. If the number of jobs exceeds the specified count, older jobs are deleted as part of housekeeping.

6.3 Configure Time-Out Parameters

The time-out parameters are configurable for each job activity. They are updated as per real node behavior by implementing individual time-out values for each activity in a job.

Note: For details on how to view and modify PIB parameters in a cloud deployment, refer to section Configuring PIB Parameters.

6.3.1 Load Control Related Configuration Parameters (MINI-LINK Indoor)

Parameter	Type	Scope	Default Value	Description
SHM_MINI_LINK_INDOOR_UPGRADEJOB_DOWNLOAD_ACTIVITY_MAX_COUNT	long	SERVICE	100	Maximum number of Download activities of UpgradeJobs that can run in parallel.
SHM_MINI_LINK_INDOOR_UPGRADEJOB_ACTIVATE_ACTIVITY_MAX_COUNT	long	SERVICE	100	Maximum number of Activate activities of UpgradeJobs that can run in parallel.
SHM_MINI_LINK_INDOOR_UPGRADEJOB_CONFIRM_ACTIVITY_MAX_COUNT	long	SERVICE	100	Maximum number of Confirm activities of UpgradeJobs that can run in parallel.

6.3.2 SHM Activities Time-out Configuration Parameters (CPP)

Table 16

Parameter	Type	Scope	Default Value	Description
SHM_CPP_BACKUPJOB_UPLOAD_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Upload activity of ERBS Backup Job. — NotNull

Parameter	Type	Scope	Default Value	Description
				— units: mins
SHM_CPP_UPGRADEJOB_INSTALL_ACTIVITY_TIME_OUT	Integer	SERVICE	480	Time-out for Install activity of ERBSUpgrade Job. — NotNull — units: mins
SHM_CPP_UPGRADEJOB_VERIFY_ACTIVITY_TIME_OUT	Integer	SERVICE	120	Time-out for Verify activity of ERBS Upgrade Job. — NotNull — units: mins
SHM_CPP_UPGRADEJOB_UPGRADE_ACTIVITY_TIME_OUT	Integer	SERVICE	120	Time-out for Upgrade activity of ERBS Upgrade Job. — NotNull — units: mins
SHM_CPP_UPGRADEJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Confirm activity of ERBS Upgrade Job. — NotNull — units: mins
SHM_CPP_RESTOREJOB_DOWNLOAD_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Download activity of ERBS Restore Job. — NotNull — units: mins
SHM_CPP_RESTOREJOB_VERIFY_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Verify activity of ERBS Restore Job — NotNull — units: mins
SHM_CPP_RESTOREJOB_INSTALL_ACTIVITY_TIME_OUT	Integer	SERVICE	480	Time-out for Install activity of ERBS Restore Job. — NotNull — units: mins
SHM_CPP_RESTOREJOB_RESTORE_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Restore activity of ERBS Restore Job. — NotNull — units: mins
SHM_CPP_RESTOREJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Confirm activity of ERBS Restore Job. — NotNull

Parameter	Type	Scope	Default Value	Description
				— units: mins
SHM_CPP_BACKUPHOUSEKEEPINGJOB_CLEANCV_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for CleanCV activity of CPP BackupHousekeeping Job. — NotNull — units: mins
SHM_CPP_DELETEUPGRADEPACKAGEJOB_DELETEUPGRADEPACKAGE_ACTIVITY_TIME_OUT	Integer	SERVICE	25	Specifies Activity Timeout on SHM Delete upgradepackage job for cpp based nodes.

6.3.3 SHM Activities Time-out Configuration Parameters (ERBS)

Parameter	Type	Scope	Default Value	Description
ERBS_BACKUPJOB_UPLOAD_ACTIVITY_TIME_OUT	Integer	SERVICE	45	Time-out for Upload activity of ERBS Backup Job. — NotNull — units: min
ERBS_UPGRADEJOB_INSTALL_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Install activity of ERBSUpgrade Job. — NotNull — units: min
ERBS_UPGRADEJOB_VERIFY_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Verify activity of ERBS Upgrade Job. — NotNull — units: min
ERBS_UPGRADEJOB_UPGRADE_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Upgrade activity of ERBS Upgrade Job. — NotNull — units: min
ERBS_UPGRADEJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Confirm activity of ERBS Upgrade Job. — NotNull — units: min
ERBS_RESTOREJOB_DOWNLOAD_ACTIVITY_TIME_OUT	Integer	SERVICE	45	Time-out for Download activity of ERBS Restore Job. — NotNull

Parameter	Type	Scope	Default Value	Description
				— units: min
ERBS_RESTOREJOB_VERIFY_ACTIVITY_TIME_OUT	Integer	SERVICE	15	Time-out for Verify activity of ERBS Restore Job — NotNull — units: min
ERBS_RESTOREJOB_INSTALL_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Install activity of ERBS Restore Job. — NotNull — units: min
ERBS_RESTOREJOB_RESTORE_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Restore activity of ERBS Restore Job. — NotNull — units: min
ERBS_RESTOREJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Confirm activity of ERBS Restore Job. — NotNull — units: min
ERBS_DELETEUPGRADEPACKAGEJOB_DELETEUPGRDAEPACKAGE_ACTIVITY_TIME_OUT	Integer	SERVICE	25	Specifies Activity time-out on SHM Delete upgradepackage job for ERBS-based nodes.

6.3.4 SHM Activities Time-out Configuration Parameters (MINI-LINK Indoor)

Parameter	Type	Scope	Default Value	Description
SHM_MINI_LINK_INDOOR_UPGRADEJOB_DOWNLOAD_ACTIVITY_TIME_OUT	Integer	SERVICE	180	Time-out for Download activity of MINI-LINK Indoor Upgrade Job. — NotNull — units: mins
SHM_MINI_LINK_INDOOR_UPGRADEJOB_ACTIVATE_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Activate activity of MINI-LINK Indoor Upgrade Job. — NotNull — units: mins
SHM_MINI_LINK_INDOOR_UPGRADEJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Confirm activity of MINI-LINK Indoor Upgrade Job.

Parameter	Type	Scope	Default Value	Description
				— NotNull — units: mins

6.3.5 SHM Activities Time-out Configuration Parameters (SGSN-MME)

Parameter	Type	Scope	Default Value	Description
SGSN_BACKUPJOB_UPLOADBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	660	Time-out for Upload activity of SGSN Backup Job.
SGSN_BACKUPJOB_CREATEBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	120	Time-out for Create backup activity of SGSN Backup Job.
SGSN_RESTOREJOB_RESTOREBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	40	Time-out for Restore Backup activity of SGSN Restore Job.
SGSN_RESTOREJOB_DOWNLOADBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	720	Time-out for Download Backup activity of SGSN Restore Job.
SGSN_UPGRADEJOB_PREPARE_ACTIVITY_TIME_OUT	Integer	SERVICE	180	Time-out for Prepare activity of SGSN Upgrade Job.
SGSN_UPGRADEJOB_VERIFY_ACTIVITY_TIME_OUT	Integer	SERVICE	5	Time-out for Verify activity of SGSN Upgrade Job.
SGSN_UPGRADEJOB_ACTIVATE_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Activate activity of SGSN Upgrade Job.
SGSN_MME_DELETEUPGRADEPACKAGEJOB_DELETEUPGRDAEPACKAGE_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Specifies Activity time-out on SHM Delete upgradepackage job for SGSN-MME nodes.

6.3.6 SHM Activities Time-out Configuration Parameters (Radio Node)

Parameter	Type	Scope	Default value	Description
RADIONODE_BACKUPJOB_CREATEBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	15	Time-out for Create backup activity of Radio Backup Job.
RADIONODE_BACKUPJOB_UPLOADBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Upload Backup activity of Radio Backup Job.
RADIONODE_RESTOREJOB_RESTOREBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	45	Time-out for Restore Backup activity of Radio Restore Job.
RADIONODE_RESTOREJOB_DOWNLOADBACKUP_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Download Backup activity of Radio Restore Job.

Parameter	Type	Scope	Default value	Description
RADIONODE_RESTOREJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	20	Time-out for Confirm activity of Radio Restore Job.
RADIONODE_UPGRADEJOB_PREPARE_ACTIVITY_TIME_OUT	Integer	SERVICE	60	Time-out for Prepare activity of Radio Upgrade Job.
RADIONODE_UPGRADEJOB_VERIFY_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Verify activity of Radio Upgrade Job.
RADIONODE_UPGRADEJOB_ACTIVATE_ACTIVITY_TIME_OUT	Integer	SERVICE	30	Time-out for Activate activity of Radio Upgrade Job.
RADIONODE_UPGRADEJOB_CONFIRM_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Confirm activity of Radio Node Upgrade Job
RADIONODE_LICENSEJOB_INSTALL_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Install Activity of Radio Node License Job
RADIONODE_DELETEUPGRADEPACKAGEJOB_DELETEUPGRADEPACKAGE_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Specifies Activity time-out on SHM Delete upgradepackage job for RADIO nodes.

6.3.7 SHM Activities Time-out Configuration Parameters (ECIM)

Table 17

Parameter	Type	Scope	Default value	Description
SHM_ECIM_DELETEUPGRADEPACKAGEJOB_DELETEUPGRADEPACKAGE_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Specifies Activity Timeout on SHM Delete upgradepackage job for ECIM nodes.

6.3.8 SHM Activities Time-out Configuration Parameters (MGW Node)

Table 18

Parameter	Type	Scope	Default value	Description
MGW_DELETEUPGRADEPACKAGEJOB_DELETEUPGRADEPACKAGE_ACTIVITY_TIME_OUT	Integer	SERVICE	25	Specifies Activity Timeout on SHM Delete upgradepack

Parameter	Type	Scope	Default value	Description
				age job for MGW nodes.

6.3.9 SHM Activities Time-out Configuration Parameters (AXE)

Parameter Name	Type	Scope	Default Value	Descripton
AXE_UPGRADE_ACTIVITY_HANDLE_TIMEOUT_IN_MINUTES	Integer	SERVICE	600	Time-out for Upgrade activity of AXE Upgrade Job, beyond which job will get failed. — Not Null — Units: mins
AXE_CREATEBACKUP_ACTIVITY_HANDLE_TIMEOUT_IN_MINUTES	Integer	SERVICE	30	Time-out for Create backup activity of AXE Backup Job, beyond which job will get failed. — Not Null — units: mins
AXE_UPLOADBACKUP_ACTIVITY_HANDLE_TIMEOUT_IN_MINUTES	Integer	SERVICE	30	Time-out for Upload backup activity of AXE Backup Job, beyond which job will get failed. — Not Null — units: mins
SHM_AXE_BACKUPJOB_CREATEBACKUP_ACTIVITY_WAIT_TIME_TO_START_POLLING_IN_MINUTES	Integer	SERVICE	20	Time-out for create backup activity to start polling of AXE nodes, beyond which job will get failed. — Not Null — units: mins
SHM_AXE_BACKUPJOB_UPLOAD_ACTIVITY_WAIT_TIME_TO_START_POLLING_IN_MINUTES	Integer	SERVICE	20	Time-out for upload backup activity to start polling of AXE nodes, beyond which job will get failed. — Not Null — units: mins
SHM_AXE_LICENSEJOB_INSTALL_ACTIVITY_TIME_OUT	Integer	SERVICE	10	Time-out for Install License activity of AXE License Job, beyond which job will be failed. — Not Null

Parameter Name	Type	Scope	Default Value	Descripton
				— units: mins
AXE_DELETEBACKUP_ACTIVITY_HANDLE_TIMEOUT_IN_MINUTES	Integer	SERVICE	10	Handle timeout for delete backup job activity of AXE nodes, beyond which job will get failed. — Not Null — units: mins

6.3.10 SHM Dependency Retry Time-out Values

Parameter	Type	Scope	Default value	Description
dpsRetryCount	int	SERVICE	30	Number of times Data Persistence Service is checked if it is unavailable.
wfsRetryCount	int	SERVICE	5	Number of times Workflow Service is checked if it is unavailable.
smrsRetryCount	int	SERVICE	5	Number of times a retry is executed for the SMRS service if it is unavailable.
identityMgmtServiceRetryCount	int	SERVICE	5	Number of times Identity Management Service is checked if it is unavailable.
dpsMoActionRetryCount	int	SERVICE	3	Number of times DPS MO Action is retried if it fails.
moActionWaitInterval_ms	int	SERVICE	10000	Amount of time to wait for Data persistence service to retry the MO Action.
wfsSubmitRetryCount	int	SERVICE	3	Number of times Workflow Service is checked for submission of workflows.
smrsImportRetryCount	int	SERVICE	3	Number of times SMRS service is checked for retry in order to import a file.
dpsWaitInterval_ms	int	SERVICE	3000	Amount of time to wait for the Data Persistence Service retry check.
wfsWaitInterval_ms	int	SERVICE	15000	Amount of time to wait for the Workflow Service retry check.
smrsWaitInterval_ms	int	SERVICE	15000	Amount of time to wait for the SMRS retry check.

Parameter	Type	Scope	Default value	Description
identityMgmtServiceWaitInterval	int	SERVICE	15	Amount of time to wait for the Identity Management Service retry check.
wfsSubmitWaitInterval_ms	int	SERVICE	5000	Amount of time wait for the Workflow Service Submission retry check.
smrsImportWaitInterval_ms	int	SERVICE	5000	Amount of time to wait for the SMRS Importing retry check.
dpsOptimisticLockWaitInterval_ms	int	SERVICE	2000	Amount of waiting time for the DPS service retry when an Optimistic Lock issue occurs.
readAttributesRetryCount	int	SERVICE	3	Number of times that the Data Persistence Service is checked for reading attributes directly from the node, if it is unavailable
readAttributesWaitInterval_ms	int	SERVICE	2000	Time to wait for reading attributes directly from node retry check.
AXE_INV_PERSISTENCE_RETRY_COUNT	int	SERVICE	30	Number of times Data Persistence Service will be checked for, if it is unavailable.
AXE_INV_PERSISTENCE_WAIT_INTERVAL_MS	int	SERVICE	3000	Amount of time to wait for the Data Persistence Service retry check.
AXE_INV_MAX_RETRY_COUNT_TO_QUERY_WINFIOL	int	SERVICE	12	AXE inventory maximum retry count to fetch inventory data from Winfiol.
AXE_INV_JMS_MSG_DELIVERY_DELAY_IN_SEC	int	SERVICE	10	AXE inventory JMS message delivery delay in seconds.

6.3.11 Workflow Configurable Parameters

Parameter	Type	Scope	Default value	Description
wfs_msg_correlation_timeout_interval_ms	Long	SERVICE	3000	Parameter for WorkflowInstance Notifier Thread.sleep().
wfs_msg_correlation_retry_count	Integer	SERVICE	10	Retry count for WorkflowInstance Notifier correlate Message loop.
wfs_mo_action_timeout_interval_min	Integer	SERVICE	20	WFS mo_action time-out interval.

Parameter	Type	Scope	Default value	Description
wfs_upgrade_timeout_interval_min	Integer	SERVICE	50	WFS upgrade time-out interval.

6.3.12 SHM Job Configurable Parameters

Parameter	Type	Scope	Default value	Description
job_query_max_batch_size	int	SERVICE	300	Change Event Attribute for SHM Jobs.
instrumentation_lock	int	SERVICE	4	Retry count for Instrumentation Lock.
SHM_PERFORM_FAILSAFE_BACKUP;	boolean	SERVICE	true	Activate or deactivate Failsafe feature for SHM License Job.

6.3.13 Misc Parameters

Parameter	Type	Scope	Default value	Description
shmInventoryQueryBatchSize	v	SERVICE	200	Batch size for inventory queries.
asyncCallsCount	Int	SERVICE	10	Count for asynchronous calls.
retryCount	Int	SERVICE	10	Number of times a service is checked if it's not found.
waitTimeInSeconds;	Int	SERVICE	30	Time to wait between each check.
corba_connection_timeout_interval_ms;	Integer	SERVICE	60000	Configurable parameter for PICIHandler CORBA network connection TIME_OUT_IN_MILLISECONDS.
SHM_INVENTORY_FOLLOW_CM_SYNC;	Boolean	SERVICE	false	Change event attribute for CM Sync.
SHM_INVENTORY_CORBA_RETRY_COUNT	Int	SERVICE	3	Number of times SHM will try to fetch the inventory XML file path by sending CORBA request.
SHM_INVENTORY_CORBA_RETRY_INTERVAL	Int	SERVICE	1000	Wait interval for next retry to fetch the inventory XML file path by sending CORBA request.
SHM_INVENTORY_FTP_RETRY_COUNT	Int	SERVICE	3	Number of times SHM will try to fetch the inventory XML file from the node through SFTP/FTP.
SHM_INVENTORY_FTP_RETRY_INTERVAL	Int	SERVICE	1000	Interval for the next retry to fetch the inventory XML

Parameter	Type	Scope	Default value	Description
				file from the node through SFTP/FTP.
DAILY_SCHEDULE_TIME_FOR_AXE_INVENTORY_SYNC_IN_HOURS	Int	SERVICE	2	Periodic Inventory Sync Start time in hours for Axe Nodes.

6.4 Controlling Failsafe Backup during Radio Node License Install

Activate the Failsafe backups function to automatically restore the system from a backup if the configuration is not confirmed within a defined time period. This can happen in case of license key file (LKF) fails during the license installation on nodes, causing OAM Connectivity issues.

The SHM_PERFORM_FAILSAFE_BACKUP configuration parameter is set to `true` by default. To disable `FailsafeBackup` during license installation, set the configuration parameter SHM_PERFORM_FAILSAFE_BACKUP to `false`.

SHM performs a failsafe backup during license installation for all ECIM nodes which include BRM fragment version 3.4.0 and above.

Note: For details on how to view and modify PIB parameters in a cloud deployment, refer to section Configuring PIB Parameters.

Table 19 Parameter Description

Parameter	Type	Scope	Default Value	Description
SHM_PERFORM_FAILSAFE_BACKUP	boolean	SERVICE	True	Activate or deactivate the Failsafe backups function to automatically restore the system from a backup if the configuration is not confirmed within a defined time period.

6.5 Controlling Inventory Supervision Flow for CPP Based Nodes

The Network Element inventory data is stored in the SHM application initially when the Network Element (NE) is added into ENM. By default, this stored inventory data doesn't get updated with the changes made on the NE. Enabling Inventory Supervision keeps the inventory data persisted in the database in sync with the changes on the NE automatically.

When Inventory Supervision is enabled, inventory changes on the node are tracked by observing the changes in the values of the below mentioned attributes of the respective Managed Objects (MOs):

Licensing	`lastLicensingPiChange`
SwManagement	`lastUpPiChange`
Equipment	`lastHwPiChange`

Any change in the attribute values of Licensing, Equipment or SoftwareManagement MOs on the node result in triggering the Inventory synchronization.

SHM Service allows the following two ways to synchronize the inventory for CPP nodes:

— Enabling Inventory Supervision based on CM supervision and Sync status. This can be achieved by setting the configuration parameter `SHM_INVENTORY_FOLLOW_CM_SYNC` to `true`.

— Enabling Inventory Supervision explicitly using ENM CLI commands.

Note: For details on how to view and modify PIB parameters in a cloud deployment, refer to section Configuring PIB Parameters.

Prerequisites

— To Enable or Disable Inventory Supervision, the NE has to be successfully added to ENM using the Add Node commands.

— A user with CM Administrator role exists on the system. Only a user with CM admin role can perform `cmedit` operations in ENM CLI.

Steps

1. To update the configuration parameter on shmcoreserv deployed service use the following table.

Parameter	Type	Scope	Default Value	Description
SHM_INVENTORY_FOLLOW_CM_SYNC	boolean	SERVICE	False	Enable or disable Inventory Supervision based

Parameter	Type	Scope	Default Value	Description
				on CM supervision and Sync status.

2. Explicitly enable Inventory Supervision by performing a manual sync:

```
cmedit action <InventoryFunction FDN> synchronize.(invType=ALL)
```

Note: Inventory Supervision is not dependent on Manual Sync changes.

Manual Inventory Synchronization is allowed though the Inventory supervision is not turned on.

Example

Explicitly call the Inventory Supervision:

```
cmedit action NetworkElement=TEST3,SHMFunction=1,InventoryFunction=1 synchro →
nize.(invType=ALL)
```

Results

If the configuration parameter `SHM_INVENTORY_FOLLOW_CM_SYNC` is set to `true`, Inventory supervision aligns with CM supervision. If the CM supervision is turned ON, the inventory data is automatically updated whenever there is change on the NE.

The inventory data in the SHM application is updated with the latest inventory on the NE in both scenarios.

6.6 Configuration parameters for Node Software Upgrade

SHM does not allow the software package to be installed on the node if it's already running with same software package. SHM checks the product number and revision of the software package to be installed through the node active software product number and revision.

When the Operator selects validation step in upgrade job creation, the nodes that are running on the same software package will be skipped during the job creation itself and information will be displayed to the user in job summary step of job creation wizard.

If the Operator has not selected the validation step as part of job execution, SHM skips the nodes running with same software package at the job execution phase. For prepare and install activities SHM checks if the package is already available on the node but not in running state and will skip the install/prepare activity.

This entire behavior can be governed using the configuration parameters. By default this configuration parameter will be true, it won't allow the same package to be installed if it is available on the node and it is in running state. For testing and other purposes we can make the flag as false.

Note: For details on how to view and modify PIB parameters in a cloud deployment, refer to section Configuring PIB Parameters.

6.6.1 List of Configuration Parameters for Modeled in SHM

Table 20 Default Values for Skip Node Software Upgrade or Install Activity

Skip Node Software Upgrade/Install Activity Related Configuration Parameters.				
Parameter	**Type**	**Scope**	**Default Value**	**Description**
SKIP_NODES_AT_SAME_SWVERSION_IN_UPGRADE_VALIDATION	boolean	SERVICE	true	If the node is already at same software level then node will be skipped from upgrade process. We can override this behavior by change this parameter value to "false". The allowed values for this parameter are: [true , false]
SKIP_INSTALL_ACTIVITY_IN_UPGRADE_JOB	boolean	SERVICE	true	If selected SW pkg is already available on node and it is not in running state then install activity will be skipped. We can override this behavior by change this parameter value to "false". The allowed values for this parameter are/: [true , false]

6.7 Mark Hung Jobs to System Canceled

> The Mark Hung jobs to System Canceled feature are triggered on a daily basis. This changes the status of the Hung Main jobs and their NE jobs and activity jobs to System Canceled, based on the configuration parameter for the maximum time limit for job execution in hours.

That means when main jobs, NE jobs, or activity jobs are running up to the maximum time limit, the jobs are considered as hung jobs and the status changes to System Canceled. The maximum time limit applies to all jobs.

The following scenarios are considered as jobs with status System Canceled.

Hung Job - The jobs (main jobs, NE Jobs and activity jobs) which are running more than configured time (default time is 48 hours) are considered as hung jobs.

— Main jobs and respective NE Jobs and activity Jobs are hung. Main jobs and respective NE Jobs and activity Jobs are considered to be System Canceled.

— Main job is not considered as System Canceled, if NE Jobs and activity jobs are running.

— Main jobs are not considered as System Canceled, if any of their NE Jobs or activity jobs are scheduled or waiting for user input.

- Main job and respective NE Jobs are not considered as System Canceled, if activity jobs are scheduled or waiting for user input.
- Main job and respective NE Jobs are not considered as System Canceled, if activity jobs are running.
- Main job is not considered as System Canceled, if some NE jobs are hung and few are running. If some of NE Jobs is hung then only hung NE jobs are marked as System Canceled; remaining NE jobs status are running and main job status remains running.

The following is the maximum job execution-time limit for each job:

Maximum time limit for job execution configuration parameters

- `MAX_TIME_LIMIT_FOR_JOB_EXECUTION_IN_HOURS` - Maximum job execution-time limit in hours for hung jobs. Default value is 48 hours.

Schedule Time Parameters

The following parameters can be configured to schedule daily housekeeping of jobs and the same scheduler is used to trigger mark hung jobs to system cancelled function:

- Daily Schedule Time of Jobs Housekeeping can be determined from the following two configurable parameters:
  - `DAILY_SCHEDULE_TIME_IN_HOURS_FOR_JOB_HOUSEKEEPING` - determines the hours of scheduled time of housekeeping.
  - `DAILY_SCHEDULE_TIME_IN_MINUTES_FOR_JOB_HOUSEKEEPING` - determines the minutes of scheduled time of housekeeping.

    For example, if the value of the first parameter is `10` and the value of the second parameter is `30`, the daily schedule time of Jobs Housekeeping is `10h 30min`.

6.7.1 Default Values to Determine Hung Jobs Parameters Table

For details on how to view and modify PIB parameters using service name shmcoreserv, refer to Configuring PIB Parameters.

Table 21 Parameters to Determine Hung Jobs

Mark Hung job Related Configuration Parameters.				
Parameter	**Type**	**Scope**	**Default Value**	**Description**
`DAILY_SCHEDULE_TIME_IN_HOURS_FOR_JOB_HOUSEKEEPING`	int	SERVICE	5	Hours of Daily Schedule Time when housekeeping is kicked off. The allowed values for this parameter are as follows:

Mark Hung job Related Configuration Parameters.				
Parameter	**Type**	**Scope**	**Default Value**	**Description**
				[Max value = 23 hours] [Min value = 0 hours]
DAILY_SCHEDULE_TIME_IN_MINUTES_FOR_JOB_HOUSEKEEPING	int	SERVICE	0	Minutes of Daily Schedule Time when housekeeping is kicked off. The allowed values for this parameter are as follows: [Max value = 59 min] [Min value = 0 min]
MAX_TIME_LIMIT_FOR_JOB_EXECUTION_IN_HOURS	int	SERVICE	48	Maximum time limit for job execution in hours for hung jobs

6.8 Configurable Parameters for SHM Inventory

By default, only software version data can be exported. To export software items also the SHM_INV_EXPORT_INCLUDE_SW_SUB_ITEMS parameter has to be set to true.

For AXE nodes, SHM_INV_EXPORT_INCLUDE_SW_SUB_ITEMS is not applicable. By default, Software Versions, Software Items and Block & Corrections data will be exported.

Note: For details on how to view and modify PIB parameters in a cloud deployment, refer to section Configuring PIB Parameters.

Parameter	Type	Scope	Default Value	Description
SHM_INV_EXPORT_INCLUDE_SW_SUB_ITEMS	boolean	SERVICE	False	Enable or disable exporting of software items.

6.9 Configurable Parameters for SHM Alarms

Steps

SHM can generate an internal alarm for SHM job failures. Internal alarms are generated to indicate any fault at network while performing network operations.

It is configurable for which job types, the alarms must be generated.

Note: For details on how to view and modify PIB parameters using service name shmserv, refer to section Configuring PIB Parameters.

6.9.1 Configuration Parameters Modeled in SHM for Alarms Raised on SHM Job Failure

Parameter	Type	Scope	Default Value	Description
SEND_ALARM_ON_BACKUP_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if alarm has to be raised on SHM Backup job failure
SEND_ALARM_ON_UPGRADE_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if alarm has to be raised on SHM Upgrade job failure
SEND_ALARM_ON_LICENSE_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if alarm has to be raised on SHM License job failure
SEND_ALARM_ON_RESTORE_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if alarm has to be raised on SHM Restore job failure
SEND_ALARM_ON_DELETEBACKUP_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if alarm has to be raised on SHM Delete backup job failure
SEND_ALARM_ON_BACKUP_HOUSEKEEPING_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if alarm has to be raised on SHM Backup housekeeping job failure
SEND_ALARM_ON_DELETE_UPGRADEPACKAGE_JOB_FAILURE	Boolean	Service	true	Configuration parameter which specifies if an alarm has to be raised on SHM Delete upgradepackage job failure.

6.10 Preserve the SSH Keys after ENM Installation

> Create backups of the Secure Shell (SSH) keys to preserve the VMs SSH keys after ENM initial installation process.

Note: The ssh key mismatch issue is observed only on L16B and L17A(LA) version nodes. If you use 16B or L17A(LA) version nodes then follow this procedure.

After completing the ENM initial installation process, the Software Hardware Manager (SHM) job fails with `FTP server not reachable` error for Evolved

RBS (ERBS) and Media Gateway (MGW) nodes. The network element is unable to connect to the smrsserv Virtual Machine (VM) because of a SSH Key mismatch. To prevent that, back up the SSH keys present on one of the SecServ VMs and restore them on the smrsserv VM.

Irrespective of the error `FTP server not reachable`, the same process is applicable for Router6672, Router6675, Router6x71, Router6273, and Router6274 node types in ENM if the Software Hardware Manager (SHM) Upgrade jobs fail due to SSH Key mismatch.

Prerequisites

— The SHM backup is successfully uploaded to the ENM Software Management distribution Repository Services (SMRS) for an ERBS or MGW node. Refer to the *ENM online help* to create the SHM Backup job and upload it to ENM.

— Access to the Management Server (MS).

— Access to the smrsserv VM.

— Access to the VM SSH keys.

Steps

1. Log on to first SecServ VM, then switch to the root user by following steps in *Connect to a Virtual Machine on an ENM on Cloud Deployment*.

2. Create directory for all smrsserv instances on the `/ericsson/tor/data`.

   Example

   ```
   [root@svc-5-secserv ~]# mkdir -p /ericsson/tor/data/vm-host-keys/svc-5-smrss →
   erv/
   [root@svc-5-secserv ~]# mkdir -p /ericsson/tor/data/vm-host-keys/svc-6-smrss →
   erv/
   ```

3. Locate the following SSH keys in the `/etc/ssh/` directory on any SecServ VM and copy them to directories created in the previous step.

   Example

   ```
   [root@svc-5-secserv ssh]# scp /etc/ssh/ssh_host_*  /ericsson/tor/data/vm-hos →
   t-keys/svc-5-smrsserv/
   [root@svc-5-secserv ssh]# scp /etc/ssh/ssh_host_*  /ericsson/tor/data/vm-hos →
   t-keys/svc-6-smrsserv/
   ```

   Note: The SSH keys must be the same on all the smrsserv instances. Make sure you copy the same SSH keys to all the directories created in Step 4.

   For every clean start of smrsserv VMs, ssh keys will be automatically copied from the directories created in Step-4 to all smrsserv instances.

Results

A backup of SSH key files is created and for every clean start of smrsserv VM, automatic restoration of SSH Keys happens on all smrsserv instances.

6.11 Update Sync Package Functionality Interval

Since VNF packages can be onboarded directly to Virtual Network Function Orchestrators (NFVOs) without using ENM SHM import and onboard features, ENM automatically synchronizes package information from configured NFVOs every 2 minutes by default. Follow this instruction to modify this time interval in a physical deployment.

Note: For details on how to view and modify PIB parameters using service name shmserv, refer to section Configuring PIB Parameters.

Parameter	Type	Scope	Default Value	Description
`nfvoSyncInterval`	int	SERVICE	2 minutes	Increase or decrease time interval to synchronizes package information from configured NFVOs.

6.12 Configurable Parameters for Instantaneous Licensing

> These parameters are configured to complete the activity within the timeout value.

6.12.1 Set Activity Time-Out Values

> Parameters are configured to complete the activity within the timeout value.

Note: For details on how to view and modify PIB parameters using service name SHMCORESERV, refer to section Configuring PIB Parameters.

Table 22 Activity Time-Out Values

Parameter	Description	Type	Default Value (in Minutes)	Min Value (in Minutes)	Max Value (in Minutes)	Note
`SHM_ECIM_LICENSE_REFRESH_JOB_REFRESH_ACTIVITY_TIME_OUT`	Time-out for refresh activity of ECIM License Request Job	Integer	`30`	-	-	

Parameter	Description	Type	Default Value (in Minutes)	Min Value (in Minutes)	Max Value (in Minutes)	Note
SHM_ECIM_LICENSE_REFRESH_JOB_REQUEST_ACTIVITY_TIME_OUT	Time-out for request activity of ECIM License Request Job	Integer	160	-	-	Value changed to align with ELIS response time-out value (120 mins).
SHM_ECIM_LICENSE_REFRESH_JOB_INSTALL_ACTIVITY_TIME_OUT	Time-out for install activity of ECIM License Request Job	Integer	5	-	-	

6.12.2 Read Configurable Parameters for Load Controller

> The parameters in this sections are configured to allow the maximum no of activities to run in parallel.

Note: For details on how to view and modify PIB parameters using service name SHMCORESERV, refer to section Configuring PIB Parameters.

Table 23 Load Controller

Parameter	Description	Type	Default Value	Min Value	Max Value	Note
SHM_ECIM_LICENSE_REFRESH_JOB_REFRESH_ACTIVITY_MAX_COUNT	Maximum number of `refresh` activities that can be run in parallel	long	1	-	-	
SHM_ECIM_LICENSE_REFRESH_JOB_REQUEST_ACTIVITY_MAX_COUNT	Maximum number of `request` activities that can be run in parallel	long	1	-	-	
SHM_ECIM_LICENSE_REFRESH_JOB_INSTALL_ACTIVITY_MAX_COUNT	Maximum number of `install` activities that can be run in parallel	long	1	-	-	

6.12.3 Read Configurable Parameters for LKF Request Polling Service

> The parameters in this section are configured for polling services.

Steps

Note: For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Table 24 LKF Request Polling Service

Parameter	Description	Type	Default Value	Min Value	Max Value	Note
autoLicenseRefreshJobPollerInterval	Poll Interval to gather LKF Requests sent from Network Elements and create a job automatically	int	2	1	2	
autoLicenseRefreshJobSchedulerInterval	Interval to create automatic install license job on completion of specified time	int	10	5	10	
enableAutoLicenseRefreshJobCreation	Flag to enable or disable automatic License Request Job	boolean	true	-	-	

6.13 Integration of Network Element Software Store (CAS-C) with ENM

After installing ENM and Network Element Software Store (CAS-C) Server, there are some Network Element Software Store connectivity parameters that have to be configured, for SHMSERV to integrate the ENM successfully with the Network Element Software Store (CAS-C). This workflow describes the steps need to be run.

6.13.1 Update Scheduler Interval Parameter for Polling to the Network Element Software Store (CAS-C)

The Network Element Software Store (CAS-C) provides the capability to change the ENM PIB Parameter NE_SOFTWARE_STORE_SCHEDULER_INTERVAL_IN_MINUTES at any time.

The Scheduler interval is used to specify the time interval for downloading the Network Element Software Package Metadata files. To poll the metadata information, this parameter needs to be updated.

The default value for this parameter is 5259600 minutes (10 years). The minimum value allowed for this parameter is 60.

Note: For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Parameter	Scope	Default Value	Description	Value Range	Recommended Value
NE_SOFTWARE_STORE_SCHEDULER_INTERVAL_IN_MINUTES	SERVICE	5259600(1)	Scheduler Interval in minutes Network Element Software Store (CAS-C) server connection.	>=10	60

(1) The default value for the scheduler interval in minutes is **5259600**. The value range is **>=10**. The recommended value when integrating Network Element Software Store (CAS-C) with ENM is **60**. Due to the nature of the operation, setting up a value lower than **60** can cause higher network congestion. Also, it may cause the scheduler to run multiple times in parallel. However, it is applicable only if AUTOMATIC_SWPKG_IMPORT_FROM_NE_SOFTWARE_STORE is set to **True**.

If there is an issue when setting up the ENM PIB parameters, refer to *Troubleshoot Setting the PIB Parameters* in the ENM Configuration Troubleshooting Guide.

6.13.2 Update Parameters for the Network Element Software Store (CAS-C)

The default configured values can be updated. The following instructions describe how to update the PIB parameters in SHMSERV VM. These values need to be provided by the operator.

6.13.2.1 Parameter Table - Configuration Management Service for the Network Element Software Store (CAS-C)

Note: For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Table 25 Configuration Management Service for the Network Element Software Store (CAS-C)

Parameter Name	Default Value	Warnings	Explanation	Value Range	Recommended Value
NE_SOFTWARE_STORE_IP_ADDRESS	0.0.0.0		IP address for Network Element Software Store (CAS-C) server connection.	Only valid IPv4 addresses are allowed	N/A
NE_SOFTWARE_STORE_PORT_NUMBER	22		Port Number for Network Element Software Store (CAS-C) server connection.	0-65535	N/A

Parameter Name	Default Value	Warnings	Explanation	Value Range	Recommended Value
NE_SOFTWARE_STORE_USERNAME	Test_User		Username for Network Element Software Store (CAS-C) server connection		N/A
NE_SOFTWARE_STORE_SCHEDULER_INTERVAL_IN_MINUTES	5259600		Scheduler Interval in minutes Network Element Software Store (CAS-C) server connection.	>=10	60
AUTOMATIC_SWPKG_IMPORT_FROM_NE_SOFTWARE_STORE	false		If automatic import from Network Element Software Store (CAS-C) is enabled	true/false	N/A
TOTAL_NUMBER_OF_IMPORTED_SOFTWARE_PACKAGES_TO_RETENTION	12		Total number of imported Software Packages to be retained for each NE type in ENM	>=1	N/A

6.13.3 Re-generate Keypair Values for Connecting ENM to the Network Element Software Store (CAS-C)

If the user wishes to update the public/private key pair used to connect ENM to the Network Element Software Store (CAS-C), this workflow describes how to overwrite the currently active key.

Prerequisites

— Root access to ENM Management Server (MS-1).

— Root login credentials for all VMs and Services of ENM physical.

— SHMSERV VM instance used for updating PIB parameters should be online.

— Verify that the public and private key files have been generated in the directory: `/ericsson/tor/data/shm/sftp/temporary_keys/`

— The Integration of Network Element Software Store (CAS-C) with ENM workflow has been run successfully.

Note: Refer to the section: *Integration of the CAS-C dropbox with ENM* of document *CAS Software Dropbox End-User Guide* 6/1553-HSC 901 110 Uen for instructions on how to integrate the Network Element Software Store (CAS-C) with ENM.

Steps

Refer to the section: *Order Access for ENM* of document *CAS Software Dropbox End-User Guide* 6/1553-HSC 901 110 Uen for instructions on how to integrate the Network Element Software Store (CAS-C) with ENM.

6.13.4 Mutual Authentication Setup between ENM and CAS-C

ENM communicates to CAS-C server for instantaneous licensing functionality. Communication is done using REST on HTTPS with mutual authentication.

Mutual authentication between ENM SHM & CAS-C can be setup in three different ways:

— CAS CA and ENM CA are independent.

— ENM PKI generates certificates for both SHM and CAS.

— ENM CA subtended to the external CA and ENM generates certificates for SHM and CAS.

The operator can choose one of the options from the preceding list.

6.13.4.1 CAS CA and ENM CA are Independent

ENM CA and CAS-C CA are independent. The following procedure explains the process of setting up mutual authentication in ENM.

The import of trusts is done manually on both sides.

ENM SHM uses certificate generated by ENM_NBI_CA to communicate with CAS-C.

ENM SHM certificate is installed automatically during ENM installation or upgrade.

Prerequisites

— ENM CLI is up and running.

— A command console is opened.

— User has valid CAS-C CA certificate shared by CAS-C team in PEM format.

Steps

1. Associating CAS CA certificate to ENM SHM Trust Profile (ENM_InstantaneousLicense_TP)
2. Providing ENM_NBI_CA certificate and CRL to CAS-C

3. Configuring CAS CRL in ENM

6.13.4.1.1 Associating CAS CA Certificate to ENM SHM Trust Profile (ENM_InstantaneousLicense_TP)

1. Import the valid CAS CA certificate into ENM.

   Get the CAS certificates to local system which need to be imported in to ENM.

2. Drag and drop that certificate file to the ENM CLI App and execute the following command.

   ```
   pkiadm extcaimport -fn file:<<CAName>>.pem -cr false --name "<<CAName>>
   ```

3. Check that the certificates are imported correctly.

   ```
   pkiadm extcalist
   ```

   Sample output is as follows:

External CA Name	Subject	Issuer	Validity	Serial Number	CRL	Auto Update CRL	Trust Profiles
<<CAName>>	CN=CAS M2M CN for INV, O=Ericsson,L=Budapest, C=HU	CN=CAS M2M CN for INV,O=Ericsson,L=Budapest, C=HU	2025-08-09 11:42:57	ab7af4cefe5f84ac	false	false	ENM_InstantaneousLicense_TP

4. Download the XML file and update it.

   ```
   pkiadm profilemgmt --export --profiletype trust --name ENM_InstantaneousLice →
   nse_TP
   ```

   In the exported trust profile XML, add the separate ExternalCA tag, and under that, add CertificateAuthority tag with corresponding CA Name used in step1.

   Example:

   ```
   <ExternalCA>
      <CertificateAuthority>
         <Name><<CAName>></Name>
   ```

```
   </CertificateAuthority>
</ExternalCA>
```

5. Drag and drop the XML file into the ENM CLI and run the following command:

```
pkiadm profilemgmt --update --xmlfile file:<update_trust_prof →
ile_xml>
```

6. Verify that the Trust Profile has been updated with External CA Certificates.

```
pkiadm profilemgmt --view --profiletype trust --name ENM_InstantaneousLicens →
e_TP
```

Sample output is as follows:

```
Trust Profile Data::
Profile Validity:
External CA: <<CAName>>
Name: ENM_InstantaneousLicense_TP
Internal CA: ENM_NBI_CA
Is Active: TRUE
Modifiable: TRUE
```

7. Restart all `shmserv` instances after making these changes.

   Note: For Extra small ENM, restart `consshm` VM.

6.13.4.1.2 Providing ENM_NBI_CA certificate and CRL to CAS-C

The certificate issued by ENM_NBI_CA can be downloaded from ENM PKI Entity Management UI.

1. From ENM launcher page, navigate to ENM PKI Entity Management.
2. Select ENM_NBI_CA from the list and navigate to Certificate Summary.
3. Download the certificate that has Status: Active in PEM format.
4. Repeat steps 2 and 3 for ENM_Infrastructure_CA and ENM_PKI_Root_CA.
5. Provide the three certificates to CAS-C.
6. To provide ENM CRL to CAS, follow steps explained in section 2.22.2 Automatic Certificate Revocation List Generation, or section 2.22.3 Download Certificate Revocation List of *ENM Public Key Infrastructure System Administrator Guide*.

Note: ENM can provide CRL information in DER format, when CRLs are retrieved from CDPS. It is recommended to use the automatic CRL generation in ENM and configure it in CAS-C. CDPS URL follows the format:

```
http://<haproxy-sb>:8093/pki-cdps?ca_name=<CA_entity>&ca_cert_seria →
lnumber=<serial_of_CA_entity>
```

6.13.4.1.3 Configuring CAS CRL in ENM

CAS CRL files can be configured in ENM using either automatic or manual procedure.

To automatically configure CRL files from External URL, follow the steps in 2.11.1.3 Auto Retrieval of Vendor Credential CRL Files from External URL of *ENM Public Key Infrastructure System Administrator Guide*.

To manually configure CRL files, follow the following command:

```
pkiadm extcaupdatecrl -fn file:<file>.crl -n <<CA_NAME>>
```

Verify whether the CAS CRLs are correctly imported into the system with the command:

```
pkiadm extcalist
```

After the CRLs are imported successfully, the CRL must have true value. When the CRLs are not properly imported, the value becomes false.

6.13.4.2 ENM PKI Generates Certificates for Both SHM and CAS

ENM provides certficates to ENM SHM and CAS-C.

This task describes the procedure to generate CAS certificate from ENM PKI signed by `ENM_EXTERNAL_ENTITY_CA` to enable mutual authentication.

Prerequisites:

- ENM CLI is up and active.
- A command console is open.
- ENM launcher is open.

6.13.4.2.1 Issue Certificate to CAS-C from ENM

CAS certificate generation with `ENM_EXTERNAL_ENTITY_CA` can be done from PKI Profile Management UI.

To generate certificate to CAS-C from ENM, the following steps need to be executed:

1. Creation of Certificate Profile:

   Create Certificate profile from PKI Profile Management with `ENM_EXTERNAL_ENTITY_CA`.

   Once PKI Profile Management has launched, select the PKI certificate Profile from the Create PKI profile drop down and fill the required fields and provide the certificate validity to create certificate profile.

   Note: SAN field shall be populated with IP address(es) if certificate is issued using hostname.

   To add SAN field with IP address - While creating Certificate Profile under capabilities tab of extension section, add the IP address from the drop-down list of **SubjectAlternateName** field.

   For generating Certificate Profile from CLI, Refer to 2.9.13 of *ENM Public Key Infrastructure System Administrator Guide*.

2. Creation of Entity Profile:

   Create Entity Profile from PKI Profile Management with the previously created Certificate profile.

   Once PKI Profile Management has launched, select the PKI Entity Profile from the Create PKI profile drop-down and fill the required fields to create a certificate profile. Then, select the certificate profile, which is created in the previous step. Entity Profile must have either **Subject** or **SubjectAltName** valid fields. After that, a valid Entity Profile is generated.

   For generating Entity Profile from CLI, Refer to 2.9.13 of *ENM Public Key Infrastructure System Administrator Guide*.

3. Creation of End Entity:

   Create End entity from PKI Entity Management with the previous entity.

   Once PKI Entity Management has launched, select the PKI End Entity from the Create PKI Entity drop-down and fill the required fields to create PKI End Entity with Publish Certificate as "ON". Then, select the Entity profile, which is created in the previous step. The selected Entity Profile values are retrieved. Then End Entity will be created.

   Note: SAN field is populated with IP address(es), if certificate is issued using hostname. Provide the IP address of CAS proxy under **SubjectAlternateName** field while creating End Entity.

   For generating End Entity from CLI, Refer to 2.15 of *ENM Public Key Infrastructure System Administrator Guide*.

4. With the previous end entity, select the end entity and click **Issue** to issue the certificate using the options as PKI generation, by including full chain and PEM format. A certificate gets downloaded, which is the end entity signed certificate for CAS with `ENM_EXTERNAL_ENTITY_CA` .

   The certificate can also be generated from CLI using the following command:

   ```
   pkiadm certmgmt EECert --generate -nocsr --entityname <<EndEntityName>> --fo →
   rmat P12 --password <<Password>>
   ```

5. Provide the downloaded certificate to CAS.

6. Follow steps explained in 2.22.2 Automatic Certificate Revocation List Generation, or section 2.22.3 Download Certificate Revocation List of *ENM Public Key Infrastructure System Administrator Guide* to provide `ENM_NBI_CA's CRL to CAS`.

6.13.4.2.2 ENM_EXTERNAL_ENTITY_CA must be Associated to the SHM Trust Profile (ENM_InstantaneousLicense_TP)

This can be done from ENM PKI Profile Management UI or with CLI commands.

Using ENM CLI:

1. Download ENM_InstantaneousLicense_TP profile using following command:

   ```
   pkiadm profilemgmt --export --profiletype trust --name ENM_In →
   stantaneousLicense_TP
   ```

   Then add the following tag in the exported trust profile XML file, add the separate Internal CA tag, under that, and add CertificateAuthority tag with corresponding CA Name.

   Example:

   ```
   <TrustCAChain>
      <IsChainRequired>true</IsChainRequired>
      <InternalCA>
         <PublishCertificatetoTDPS>false</PublishCertificatetoTD →
   PS>
         <CertificateAuthority>
            <Id>22</Id>
            <Name>ENM_External_Entity_CA</Name>
            <IsRootCA>false</IsRootCA>
            <CAStatus>NEW</CAStatus>
            <PublishToCDPS>false</PublishToCDPS>
            <IsIssuerExternalCA>false</IsIssuerExternalCA>
         </CertificateAuthority>
   ```

```
   </InternalCA>
</TrustCAChain>
```

2. Drag and drop the XML file into the ENM CLI and run the following command:

```
pkiadm profilemgmt --update --xmlfile file:<update_trust_profile_xml>
```

3. Verify that the Trust Profile is updated with Internal CA Certificates.

```
pkiadm profilemgmt --view --profiletype trust --name ENM_InstantaneousLicens →
e_TP
```

   Output must contain the ENM_External_Entity_CA under Internal CA as shown in the following:

```
Trust Profile Data:
External CA:
Profile Validity:
Internal CA: ENM_External_Entity_CA, ENM_NBI_CA
Name: ENM_InstantaneousLicense_TP
Is Active: TRUE
Modifiable: TRUE
```

4. Restart all shmserv instances after making these changes.

   Note: For Extra small ENM restart, `consshm` VM.

Using ENM KPI Profile Management UI:

1. In ENM PKI Profile Management, search and edit ENM_InstantaneousLicense_TP.
2. Under Trusted CA, click on "+" icon to add a new entry.
3. Select ENM_External_Entity_CA in the dropdown and check Include full chain option.
4. Save the changes and restart all shmserv instances after making these changes.

   Note: For Extra small ENM, restart `consshm` VM.

6.13.4.3 ENM CA Subtended to the External CA and ENM Generates Certificates for SHM and CAS

Make ENM CA subtended to external CA by following the procedure explained in section 2.24 External CA Support of ENM Public Key Infrastructure System Administrator Guide.

For generating certificates for SHM and CAS, follow the steps of section ENM PKI Generates Certificates for Both SHM and CAS on page 71, once subtending is completed.

Note: Signed certificiate for ENM_PKI_Root shall not contain the **EnhancedkeyUsage** (for example, securemail), **AuthorityInformationAccess**, and **SubjectAlternateName** fields.

6.14 Configure Automatic Import of Software Packages from Network Element Software Store (CAS-C)

This section describes how to enable and configure Automatic Import of Software Packages from Network Element Software Store (CAS-C).

Prerequisites

— The Integration of Network Element Software Store (CAS-C) with ENM workflow has been run successfully.

Note: For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Steps

Parameter	Scope	Default Value	Explanation	Value Range	Recommended Value
TOTAL_NUMBER_OF_IMPORTED_SOFTWARE_PACKAGES_TO_RETENTION	SERVICE	12	Number of software packages to be retained.	>=1	-
AUTOMATIC_SWPKG_IMPORT_FROM_NE_SOFTWARE_STORE	SERVICE	False	Enable Automatic Import of Software Packages from Network Element Software Store	True / False	False
SOFTWARE_STORE_SCHEDULER_INTERVAL_IN_MINUTES	SERVICE	-	Update the scheduler interval to immediately trigger ENM polling. Follow the steps outlined in *Update Parameters for the Network Element Software Store (CAS-C)*.	-	-

6.15 Activation of Instantaneous Licensing in ENM

After installing ENM and CAS-C Server, some Instantaneous Licensing connectivity parameters must be configured for SHMSERV to integrate ENM

successfully with CAS-C. This workflow describes the steps to run to complete this.

6.15.1 Enable the Instantaneous Licensing Feature for Polling LKF Files

Steps

To enable the Instantaneous Licensing feature for polling of CAS-C Software Dropbox, change the value for the ENM PIB parameter `INSTANTANEOUS_LICENSE_ENABLED` from its default value.

In order to poll the files, this PIB parameter must be enabled. The "`true`" value enables the feature. The default value for this parameter is "`false`".

Note:
- For details on how to view and modify PIB parameters using service name SHMSERV,
- If this is the first time the feature is enabled, the scheduler runs with the default value of 30 minutes. To change the default value, see *Update Scheduler Interval Parameter*.

Parameter	Scope	Default Value	Description	Value Range	Recommended Value
INSTANTANEOUS_LICENSE_ENABLED	SERVICE	False	Enable or Disable Instantaneous Licensing feature for polling of CAS-C Software Dropbox.	True/ False	-

6.15.2 Update Scheduler Interval Parameter

To update scheduler interval set the ENM PIB parameter `INSTANTANEOUS_LICENSE_SCHEDULER_INTERVAL_IN_MINUTES`. The scheduler interval is used to specify the time interval for downloading the license files. The default value for this parameter is 30 minutes. The minimum value allowed for this parameter is 15 minutes.

Note: If there is an issue when setting up the ENM PIB parameters, see *Troubleshoot Setting the PIB Parameters on ENM* in the ENM Configuration Troubleshooting Guide [40].

6.15.2.1 Configuration Management Service for the CAS-C Software Dropbox Parameter Table

For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Table 26 Configuration Management Service for the CAS-C Software Dropbox Parameters

Parameter Name	Default Value	Description	Value Range	Component Responsible	Customer Configuration Required
INSTANTANEOUS_LICENSE_SENTINEL_PROXY_IP_ADDRESS	0.0.0.0	CAS-C Sentinel proxy IP Address for LKF REST request to ELIS.	Only valid IPv4 addresses are allowed.	CAS	Mandatory
INSTANTANEOUS_LICENSE_SOFTWARE_SUPPLY_ENDPOINT	Sentinel_URL	URL for Sentinel proxy server connection.	Valid URL.	CAS	Mandatory
INSTANTANEOUS_LICENSE_SENTINEL_PROXY_PORT_NUMBER	10010	Port Number for Sentinel proxy server connection.	0-65535	CAS	Mandatory
INSTANTANEOUS_LICENSE_BATCH_REQUEST_INTERVAL_IN_MINUTES	15	Timer interval duration for sending a request to ELIS.	1–120	ENM	Optional
INSTANTANEOUS_LICENSE_MAX_NUMBER_OF_REQUEST_PER_BATCH	1000	Maximum number of request per batch.	1–1000	ENM	Optional
INSTANTANEOUS_LICENSE_SOFTWARE_DROPBOX_ID	0	Software dropbox ID for instantaneous licensing.	>=0	CAS	Mandatory
INSTANTANEOUS_LICENSE_GLOBAL_CUSTOMER_ID	0	Global customer ID for instantaneous licensing.	>=0	CAS	Mandatory
INSTANTANEOUS_LICENSE_ENABLED	0	Enable/disable polling of LKF files from (CAS-C) Software Dropbox for the instantaneous licensing . The "true" value enables polling the LKF files and "false" disables it. The default value for this parameter is "false". In order to poll the files, the value of this PIB parameter should be changed to "true".	true/false	ENM	Mandatory
INSTANTANEOUS_LICENSE_SOFTWARE_DROPBOX_DIRECTORY_PATH	/Store/LicenseFiles	Path in CAS-C software dropbox at which the delivered license packages are made available.	Valid directory	CAS	Optional
INSTANTANEOUS_LICENSE_SCHE	30	Polling interval from ENM to	15–60	ENM	Optional

Parameter Name	Default Value	Description	Value Range	Component Responsible	Customer Configuration Required
DULER_INTERVAL_IN_MINUTES		CAS-C Software Dropbox for the license package delivered by ELIS in minutes.			
INSTANTANEOUS_LICENSE_SCHEDULER_TIMEOUT_IN_MINUTES	120	Time in minutes after which a request will be marked as timed out and will not seek delivery.	60-1440	ENM	Optional

6.15.3 Disable the Instantaneous Licensing Feature for Polling LKF Files

To disable the Instantaneous Licensing feature for polling the CAS-C Software Dropbox, change the value for the ENM PIB parameter `INSTANTANEOUS_LICENSE_ENABLED` to "`false`". This disables polling the files from CAS-C.

ENM can still send a request to ELIS to generate the license files but cannot fetch the generated files from CAS-C store.

Note:
- If there is an issue when setting up the ENM PIB parameters, see *Troubleshoot Setting the PIB Parameters* in the ENM Configuration Troubleshooting Guide.
- For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Table 27 Configuration Management Service for the CAS-C Software Dropbox Parameters

Parameter	Scope	Default Value	Description	Value Range	Recommended Value
INSTANTANEOUS_LICENSE_ENABLED	SERVICE	False	Disable the Instantaneous Licensing feature for polling of CAS-C Software Dropbox.	True/ False	-

Disable auto LKF refresh job creation when LKF refresh is triggered on the node. Refer to section *Read Configurable Parameters for LKF Request Polling Service* and modify parameter `enableAutoLicenseRefreshJobCreation` to disable automatic creation of jobs.

6.16 Configure Scheduler for Identifying Reported Stolen Equipment

SHM searches for ERS equipment that has been reported stolen, and raises an alarm on the corresponding node. The Alarm is visible in the `Alarm Monitor` on the FM NBI.

`Search for Stolen Equipment` runs daily at `10` AM.

Schedule Time Parameters

Daily Schedule Time of Stolen Equipment reporter can be determined from the following configurable parameter:

table

Parameter	Description
`dailyScheduleForStolenScrappedEquipmentTimer`	Determines the hours and minutes (HH:MM) of scheduled time of Stolen Equipment reporter.

For example, if the value of the parameter is `10:30`, the daily schedule time of Stolen Equipment reporter is `10`h `30`min.

6.17 Configuration Parameters for Stolen Equipment in SHM

All the parameters listed in Table 28 are configurable.

Note: For details on how to view and modify PIB parameters using service name SHMSERV, refer to section Configuring PIB Parameters.

Table 28 Configuration Parameters for Stolen Equipment in SHM

Parameter	Type	Scope	Default Value	Description
`dailyScheduleForStolenScrappedEquipmentTimer`	String	SERVICE	`10:00`	Time in HH:MM when Stolen Equipment Timer is kicked off every day, where HH is hours in 24 hours format [Min value = `0` hours][Max value = 23 hours] and MM is minutes [Min value = `0` minutes] [Max value = 59 minutes].

7 AMOS and Element Manager Administration Tasks

This section describes the operation and maintenance tasks for the administration of AMOS and Element Manager applications.

7.1 AMOS Adjust Logs Housekeeping

Housekeeping of the large log files generated by AMOS is a trade-off between retention time and available disk space. If housekeeping for these log files is not sufficient for a local usage profile, run this task to adjust it..

Prerequisites

— Cron is correctly installed on the Linux environment on which AMOS is running.

— `tmpwatch` is correctly installed on the Linux environment on which AMOS is running.

Steps

1. Follow the instructions in Connect to a Service on page 2, then log on to the scripting VMs.
2. Edit the `crontab` file or create it, if it does not exist already:

   ```
   [root@scp-X-scripting /] crontab -e
   ```

   The `crontab` file should be opened with your default text editor.
3. Add the following lines to the file:

   ```
   0  */8  *   *   *   /usr/bin/tmpwatch -am  8 /ericsson/log/amos/moshell_logf →
   iles/*/logs_moshell/
   0  */8  *   *   *   /usr/bin/tmpwatch -am  8 /ericsson/log/amos/moshell_logf →
   iles/*/logs_mobatch/
   ```

   As `/ericsson/log/amos` is a shared file system, for scheduling cron jobs in failover mode refer to *Reschedule Fail Over Cron Services* in the ENM Operators Guide [35].
4. If `tmpwatch` is not installed in the default `/usr/bin/` directory, check the available installation by running the following command, and changing the cron configuration line accordingly.

```
[root@scp-X-scripting /] whereis tmpwatch
```

5. To tune the cron configuration, change all occurrences of the number '8' in step 3 to the number of hours that the log files must survive.
6. Save the crontab file.

Results

All log files are deleted after a set amount of time, if not exported and archived from the AMOS working directory.

7.2 Schedule Execution of User Scripts through Cron Service

> The General Scripting Virtual Machine (VM) enables an ENM-authenticated user to schedule scripts using the cron service. The Cron service can be executed at any time.

Prerequisites

— Knowledge of Linux operation and Cron schedule.

— ENM username and password - be aware that ssh usernames are case-sensitive.

— Scripting_Operator role.

— Load must be balanced by the user across the available General Scripting VMs to not overload the machine.

Steps

1. Follow the instructions in Connect to a Service on page 2, then log on to the scripting VM to schedule a job.

   Note: Use username: `Scripting3`.

2. Create the script file to be executed.

   ```
   [scripting3@scp-1-scripting ~]$ ls -l
   total 1
   -rw-r--r--. 1 scripting3 enm_users 53 Oct 18 12:46 scripting3.sh
   ```

   Note: This example shows the user `scripting3` home folder on the `scp-1-scripting` VM, with a script file `scripting3.sh`.

3. Using the standard crontab tool, open the "vi" editor to create or edit a cron job.

```
[scripting3@scp-1-scripting ~]$ crontab -e
```

Example

```
05,20,35,50 * * * * /home/shared/scripting3/scripting3.sh
```

Note: In this example, the `scripting3.sh` created previously is executed at 05, 20, 35 and 50 minutes of every hour, every day.

4. List the cron job as follows:

```
[scripting3@scp-1-scripting ~]$ crontab -l
```

5. Once editing is complete type `:wq` to save and quit. The output should be:

```
[scripting3@scp-1-scripting ~]$ crontab -e
crontab: installing new crontab
[scripting3@scp-1-scripting ~]$ crontab -l
05,20,35,50 * * * * /home/shared/scripting3/scripting3.sh
[scripting3@scp-1-scripting ~]$
```

This schedules the cron job in the local VM, and saves a copy of that cron file in the SFS, under `/ericsson/vmcrons/$HOSTNAME/`. This can be recovered after an upgrade, or in case of a failure in the VM.

Note: Only root user can see that file and path.

```
[scripting3@scp-1-scripting ~]$ ls -ltras /ericsson/vmcrons/scp-1-scripting/
ls: cannot open directory /ericsson/vmcrons/scp-1-scripting/: Permission den →
ied
[root@scp-1-scripting ~]# ls -ltras /ericsson/vmcrons/scp-1-scripting/
total 5
0 drwxr-xr-x. 5 root          root        96 Oct 14 14:45 ..
1 -rw-------. 1 administrator enm_users   40 Oct 14 16:25 administrator
1 -rw-------. 1 scripting2    enm_users   40 Oct 14 16:38 scripting2
1 -rw-------. 1 scripting1    enm_users   40 Oct 14 16:54 scripting1
0 -rw-------. 1 amos_only     enm_users    0 Oct 17 09:22 amos_only
1 -rw-------. 1 scripting3    enm_users   58 Oct 18 13:48 scripting3
1 drw-------. 2 root          root      1024 Oct 18 13:48 .
[root@scp-1-scripting ~]# cat /ericsson/vmcrons/scp-1-scripting/scripting3
05,20,35,50 * * * * /home/shared/scripting3/scripting3.sh
```

This prevents users from editing files while the cron job is executing them, and possibly causing other issues in the system.

Once the `crontab -e` has been performed a "vi" editor is opened, and the users can add their entries.

7.2.1 Known Limitations

— If a user has the scripting_role removed, this change may take up to 10 minutes to take effect. During that time the user is still able to use crontab.

— Files remain on SFS even if the user is removed, but they are not run until the user is recreated again.

— Crontab does not generate an SSO token. Scripts running with crontab use the SSO token cached in `/home/shared/<username>/.enm_login`, which expires, by default, `10` hours after the user logs out. For this reason, cron scripts that access ENM services must not be scheduled for later than `10` hours after the user logs out.

Note: It is possible to change the default settings, either by changing them for all users through the System Security Configuration application, or by changing them for selected user(s). When changing the sessions settings for selected user(s) it is possible to create very long sessions. Contact Security Administrator if such configuration is needed.

— Cron uses session tokens to authenticate the user that scheduled a particular job. Authentication is based on the assumption that SSO keeps all active sessions. This may not always be the case. There are instances where SSO cannot keep all the sessions active, (refer to *Limitations* section in the ENM Security System Administrator Guide). In such instances, the tokens need to be recreated manually. This is done by logging in to the scripting VM.

7.3 Update AMOS Load Balancing Parameters

> AMOS uses the High Availability Proxy (HAProxy) component to distribute the load of AMOS user sessions among the available AMOS Virtual Machines (VMs).

The root user can modify the default values using the Platform Integration Bridge (PIB) script to adjust the load according to the availability of AMOS VMs.

For details on how to view and modify PIB parameters using the service name `terminal-websocket` and the app_server_address `<service VM hostname>`, refer to section Configuring PIB Parameters.

To change settings, choose the parameter name and value based on the details in section Supported Parameters for AMOS Load Balancing. To execute this operation, basic understanding of Unix Cron expressions is required.

7.3.1 Supported Parameters for AMOS Load Balancing

Attribute	Description	Default Value	Expected Output
`maxAmosSessions`	Number of maximum AMOS sessions per VM	`120`	`140`
`maxCpuUsage`	Max percentage of CPU usage	`90`	`80`
`maxMemory`	Max percentage of memory usage	`90`	`80`

Example 4

```
/ericsson/pib-scripts/etc/config.py update --name=maxAmosSession →
s --value=140 --service_identifier=terminal-websocket --app_serv →
er_address=svc-1-cmserv:8080
```

7.4 Download Certificates for AMOS SL2 Operation

Download a certificate to use AMOS on a Security Level 2 (SL2) node where the certificate has expired, or becomes invalid.

Prerequisites

— ENM user with AMOS role (Amos_Administrator or Amos_Operator).

— To execute this instruction a Certificate Entity is needed. Refer to the ENM Security System Administration Guide in the Reference List on page 298 to learn more about it.

Steps

1. From the ENM launcher, select and launch the Shell Terminal.
2. Download a certificate by running the following command:

   ```
   ssucredentials.sh
   ```

   Note: Users of AMOS via the cron scripting service can include this command in their scripts.

3. Use the `exit` command to close Shell Terminal:

   ```
   Ctrl-D
   ```

7.5 Configure AMOS Using Global Moshellrc File

The `moshellrc` is a global configuration file that contains configuration parameters for all AMOS users.

To configure AMOS across multiple VMs, the `moshellrc` is stored in a shared file system, in the `/home/shared/common/moshell/moshellrc` directory.

This task explains how to modify this configuration file.

Run this task when you want to configure AMOS for all users.

Prerequisites

— Access to all AMOS VMs.

— Access to all Scripting VMs.

— Only users in `amos_users` POSIX group have "Read" permission to the `moshellrc` file

Steps

1. Follow the instructions in Connect to a Service on page 2, then log on to the AMOS and scripting VMs.
2. Edit `moshellrc` file to access and modify the desired configuration:

```
[root@scp-1-amos ]$ vim /home/shared/common/moshell/moshellrc
```

Note: Refer to the *MoShell User Guide*, found in `/opt/ericsson/amos/moshell/UserGuide.pdf` on the ENM file system for guidance on AMOS configuration

3. If Port 22 is blocked in the ENM environment, then perform the following actions:

— Login to Linux Management Server (LMS) for physical deployments/ Login to VNF-LAF for cloud deployment as appropriate.

— Log in to the AMOS Virtual Machine.

— Read the values of `smrs_sftp_securePort` and `smrs_sftp_port_partition_enable` with the following command:

- For physical environment execute the following command:

```
cat /ericsson/tor/data/global.properties | grep -e smrs_sftp_secure
Port -e smrs_sftp_port_partition_enable
```

- For cloud environment execute the following command as root:

```
/usr/bin/consul kv get -recurse | grep -i -e smrs_sftp_securePort -
e smrs_sftp_port_partition_enable
```

— If the value of `smrs_sftp_securePort` is other than Port 22, and the value of `smrs_sftp_port_partition_enable` is true, then perform the following steps.

a. Login to any AMOS vm as root user.

b. Open the mosdhellrc file (/home/shared/common/moshell/moshellrc)

i. If "export_port" property is present then change the value of "export_port" property as below.

ii. if "export_port" property is not present then append the value of "export_port" property as below.

```
export_port=<value of smrs_sftp_securePort>
```

4. If the port 22 is enabled (not blocked) after upgrade , then make sure that the `export_port` property entry is removed from the moshellrc file (`/home/shared/common/moshell/moshellrc`).

7.6 Configuration Management Metrics for AMOS

ENM System Monitor (ESM) monitors the Configuration Management (CM) application metrics. You can monitor, set threshold, and create ESM alerts for each of the metrics. With the help of a stateful alarm, the created AMOS alerts provide information about the respective metric in ENM CM.

Requirements to Access ESM

— Access to the ENM Launcher.

— Appropriate ID, account, and permissions to access the ESM application.

Table 29 Details of the FM Metrics for AMOS

Display Name of the Metric	Description of the Metric
Current CPU usage of the virtual machine	This metric is for current CPU usage of the virtual machine
Current memory usage of the virtual machine	This metric is for current memory usage of the virtual machine
Current number of AMOS sessions running in the virtual machine	This metric is for current number of AMOS sessions running in the virtual machine

7.7 Add SSL Certificates for Domain Proxy Authentication

Install Citizens Broadband Radio Service (CBRS) Public Key Infrastructure (PKI) certificates on the Domain Proxy to enable mutual authentication with the Spectrum Access System (SAS).

Prerequisites

— PKI knowledge.

— A Domain Proxy Intermediate Certificate Authority (CA) has signed your Secure Sockets Layer (SSL) certificate.

— A signed server certificate and corresponding trust certificates are stored in a single Privacy Enhanced Mail (PEM) file.

— The servers' corresponding private key is stored in PEM format.

— The CBRS PKI Root Certificate Authorities public key certificate is stored in PEM format.

— Root access to MS (for physical ENM deployment).

— Super user privileges (for cloud ENM deployment).

— Access to the private key file for authentication (for cloud ENM deployment).

Steps

1. Establish the environment you are working on, then log on to an ENM deployment.

    a. For physical deployments, log on to the ENM MS as the litp-admin user and switch to the root user:

    If password authentication is disabled for the litp-admin user, then refer to *Log On to the MS when Password Authentication is Disabled* in the *LITP Node Hardening Instructions*.

    ```
    ssh litp-admin@ms-1
    su -
    ```

    b. For Cloud deployment:

        i. Copy the private key to the Ericsson Management Portal (EMP).

        ```
        > scp -i <cloud-user private key> <cloud-user private key> clou →
        d-user@<EMP IP Address>:/var/tmp/vm_private_key
        ```

        ii. Log on to EMP server.

        ```
        > ssh -i <cloud-user private key> cloud-user@<EMP IP Address>
        ```

2. Transfer the SSL certificate files containing the Domain Proxy certificate chain, Domain Proxy private key, and Root CA's public key to a known location on the ENM deployment.

```
[root@ms-1 ~]# scp <username>@<remotehost>:<filepath>/<certificate_chain> \
> <username>@<remotehost>:<filepath>/<private_key> \
> <username>@<remotehost>:<filepath>/<root_ca_public_key> /var/tmp
```

3. Execute the following script to add certificates to the domain proxy.

```
[root@ms-1 bin]# /ericsson/tor/data/domainProxy/bin/configure_ssl_store.sh -
c <certificate_chain> -k <private_key> -r <root_ca_public_key>
```

4. Remove the SSL certificates files from the ENM deployment.

```
[root@ms-1 bin]# rm -f /var/tmp/<certificate_chain> /var/tmp/<private_key> /
var/tmp/<root_ca_public_key>
```

7.8 Configure Timeout on SSH Sessions

> Configure the timeout on SSH sessions for AMOS and Scripting VMs to minimize security risks of SSH sessions left idle. By default, no timeout is defined.

Note: Only new sessions (opened using SSH or launched using ENM launcher) are affected by this change. However, if the timeout property is set to a value higher than the *Session Idle Time* value defined in the *ENM Security Parameters List*, all the idle sessions running through ENM launcher will get timed out as per 'Session Idle Time' value and timeout property value is ignored. Idle timeout for all webscocket applications (e.g. amos, cendio thinlinc sessions) is static configuration and set to value of 60 minutes in haproxy config. This value is not configurable via UI.

Prerequisites

Amos_Administrator or Amos_Operator role.

Scripting_Operator role.

Steps

1. Edit the configuration file to access its properties:

```
[root@scp-1-amos ]$ vim /home/shared/common/sshd_config.properties
```

2. Update the timeout property to the desired value (in seconds):

Example

```
SSH_CLIENT_SESSION_TIMEOUT=3
```

Result: It takes up to 5 minutes for the configured value to take effect.

7.9 Configure Timeout on Cendio Thinlinc Sessions

> Configure the timeout on Cendio Thinlinc Sessions for Element Manager VMs to manage sessions automatically. By default, the automatic session management is disabled.

Note: Only new Cendio Thinlinc sessions, launched using ENM launcher, are affected by this change. However, if the timeout property is set to a value higher than the *Session Idle Time* value defined in the *ENM Security Parameters List*, all the idle sessions running through ENM launcher will get timed out as per 'Session Idle Time' value and timeout property value is ignored. Idle timeout for all webscocket applications (e.g. amos, cendio thinlinc sessions) is static configuration and set to value of 60 minutes in haproxy config. This value is not configurable via UI.

Prerequisites

— Root access to the Management Server (for Physical ENM deployment).

— Root and cloud-user access to the vnflaf-services VM (for Cloud ENM deployment).

Steps

1. Edit the configuration file to access its properties:

```
[root@scp-1-elementmanager ~]# vim /home/shared/common/rdesktop/screensaver.
ini
```

2. Update the `IdleTime` and `CountdownTime` properties to the desired values, and set the flag `Enabled`.

   Example

```
[DEFAULT]
Enabled=True ;enable/disable screensaver, valid values are True or False
CountdownTime=30 ;countdown time in seconds, when screensaver appears on use
rs desktop
IdleTime=60 ;for how long user is allowed to be idle, in seconds, before cou
ntdown appears
```

Results

This change takes effect immediately and applies to the new Cendio Thinlinc sessions.

7.10 MOShell Logs Archiving Process

The AMOS file system on the NAS does not archive old logs. The number of files will increase to the point where AMOS may cause extreme performance degradation during ENM backups. The archive_moshell_logs.sh script takes care of this.

Prerequisites

— Root access to MS (for physical ENM deployment).

— Super user privileges (for cloud ENM deployment).

Expected Result

A gzip file containing logs for each user and for the given time frame will be created in the current directory, and the original log files will be deleted.

Steps

1. Follow the instructions in Connect to a Service on page 2, then log on to the scripting VM as root user.

2. Running the script `/opt/ericsson/amos/scripts/archive_moshell_logs.sh` with no arguments/options outputs the help:

   Example:

```
[root@scp-X-scripting /]$ /opt/ericsson/amos/scripts/archive_moshell_logs.sh
Archive: Backup files in the AMOS file system
_____________________________________________________________
Usage:
./archive_moshell_logs.sh LOGS_PATH [options]
Options:
-h, --help                    Display this help message
-d, --days DAYS               Files that have not been modified more than DAYS days ago will be ar →
chived [default: 3]
[root@scp-X-scripting /]$
```

3. Run the script by specifying the path to the logs directory and number of days.

   When script run is complete, user will see a gzip created for each user that has logs. Users may move these gzip files to any location that they wish. For output of script, user can check `/var/log/messages`.

   Example:

Running the following script will create gzip file under the current working folder, and remove log files that are 5 or more days old from the logs path provided.

```
[root@scp-X-scripting ]$ ls -l /ericsson/log/amos/moshell_logfiles/
total 0
drwxrwxrwt. 4 administrator enm_users 96 Apr 10 12:00 administrator
drwxrwxrwt. 4 amosop1       enm_users 96 Apr 10 11:57 amosop1
drwxrwxrwt. 4 amosop2       enm_users 96 Apr 10 11:57 amosop2

[root@scp-X-scripting ]$ /opt/ericsson/amos/scripts/archive_moshell_logs.sh /ericsson/log/amos/mo →
shell_logfiles/  --days 5
/bin/tar: Removing leading `/' from member names
/ericsson/log/amos/moshell_logfiles//amosop1/test-amosop1-log.log
/bin/tar: Removing leading `/' from member names
/ericsson/log/amos/moshell_logfiles//amosop2/test-amosop2-log.log

[root@scp-X-scripting ]$ ls -lt
total 2
-rw-r--r--. 1 root root   148 Apr 12 13:32 amosop2-amos-logs-1523536343.tar.gz
-rw-r--r--. 1 root root   148 Apr 12 13:32 amosop1-amos-logs-1523536343.tar.gz
[root@scp-X-scripting ]$
```

4. The previous script can also be scheduled to run through Cron Service, please refer to Schedule Execution of User Scripts through Cron Service on page 81.

   Example:

```
0 0 */3 * * /opt/ericsson/amos/scripts/archive_moshell_logs.sh /ericsson/log/amos/moshell_logfiles →
/  --days 5
```

7.11 Configure AMOS TBAC Parameter

This section describes how to configure AMOS TBAC setting using PIB parameter `amosTbacEnabled`.

Note: For details on how to view and modify PIB parameters using service name cmserv, refer to section Configuring PIB Parameters.

Table 30 Parameters to Enable AMOS TBAC

Parameter Name	Default Value	Description	Value Range
amosTbacEnabled	true	Enable or Disable AMOS TBAC check.	true, false

For more information about how to configure TBAC for a user, refer to *Targets and Target Groups for AMOS, Element Manager, and Cabinet Viewer* in the ENM Identity and Access Management System Administrator Guide .

7.12 Copy and Paste Long Lines in AMOS and General Scripting Shell Terminal

This topic describes the behavior of copy and paste functionality in AMOS or Scripting UI shell terminal.

— User can copy a long command of multiple lines from either the terminal or an external editor and run it as a single command. If the command is to be run as a single command, then user must use two exclamations ! ! before pasting the command into AMOS or Scripting shell terminal.

— User can copy and paste the multiple lines commands directly from the terminal or external text editor into AMOS or Scripting shell terminal. If the commands are to be run as an individual command, then, do not use two exclamations ! ! but copy and paste them into terminal.

Note: A combination of the preceding two scenarios is not supported.

8 Network Discovery Administration Tasks

This section contains the routine operation and maintenance tasks related to the administration of Network Discovery applications.

8.1 Configure Global SNMP Parameters for Node Discovery Jobs

SNMP protocol is used during network discovery operation for communicating with network elements. `Snmp_timeout and Snmp_retries` are global configuration parameters which can be configured based on the network requirements.

For details on how to view and modify SNMP parameters, refer to section Configuring PIB Parameters.

Table 31 Parameters to Configure Global SNMP for Node Discovery Jobs

Config Parameter	Default Value	Max. Value	Min. Value
`Snmp_retries`	2	3	1
`Snmp_timeout`	3	5	1

Note:

Whenever any Service Group instance is restarted, the configured parameters are reset to their respective default configurations.

9 NR-NSA Systems Topology

NR-NSA is a system operated script that enables the creation and maintenance of the NR-NSA Custom Topology and the following system-defined collections: LTE-ERBS, LTE-RadioNode and NR-RadioNode.

NR-NSA is installed to the ENM scripting cluster during the ENM deployment procedures.

The NR-NSA script is scheduled to run every night at midnight. This is configurable.

The NR-NSA script provides a Topology that depicts the relationships between the supported node types outlined in the Network Impact Report for Topology Relationships for the Discovery of NR-NSA Topology based on X2 relationships.

The NR-NSA script will also create three system-defined collections, LTERadioNode, LTE-ERBS, and NR-RadioNode, provided there are eNodeB Baseband Radio nodes, eNodeB DU Radio nodes, and gNodeB Baseband Radio nodes.

Note: It is important to note that the NR-NSA maintains itself. If a user adds collections underneath the NR-NSA Topology, they will be removed from the Topology during the next occurrence of the Script. The same happens to system-defined collections. If a user adds a collection with the same name as any of the system-defined collections, they will be removed during the next execution of the script.

The following tasks are detailed in this section:

9.1 Custom Role Setup

For the NR-NSA System Topology and the system-defined collections to be created, a new custom role is needed.

Steps

1. Log on as an ENM administrative privilege user.
2. Launch the ENM Role Management Interface.
3. Create a custom role with the following resources:
   - System_created_object.
     - Create
     - Delete

   You should now have created a custom user role.

9.2 Predefined User Setup

To use the NR-NSA System Topology functionality, an ENM user with relevant rights must be created.

Prerequisites

You have ENM administrative roles, such as create user and assign rights.

Steps

1. Log on as an ENM administrative privilege user.
2. Launch the ENM User Management Interface.
3. Create an ENM user that is assigned the following rights:
   - Scripting_Operator.
   - Network_Explorer_Administrator.
   - Cmedit_Administrator.
   - Custom role. See Custom Role Setup on page 94 for details.
4. Log off ENM User Management.

   You will have created a predefined ENM user, that will be used to run the NR-NSA system topology script.

9.3 Enable Scheduled NR-NSA Topology

This task allows you to enable the cron job that will run the NR-NSA topology at regular intervals.

Prerequisites

Predefined ENM user, created with the roles outlined in the Predefined User Setup on page 95.

Steps

1. Log on, via SSH, to ENM General Scripting VM:

```
[litp-ms] $ ssh <predefined-user>@scp-1-scripting
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run

```
su - <predefined-user>
```

as this bypasses the required Pam authentication.

2. As the <predefined-user>, execute the setup.py script, then follow prompts:

```
[scp-1-scripting~] $ cd /opt/ericsson/nr-nsa-systems-topology/bin/
[scp-1-scripting~] $ ./setup.py
```

Note: Make sure you use the same password as in Step 1 because password validation against the LDAP/PAM does not take place during Step 2.

3. Log off from the General Scripting VM.

   The generated cron job becomes active after 30 minutes.

   The NR-NSA Systems Topology has now been enabled. By default, daily at midnight, the script will run and perform updates to the NR-NSA Custom Topology and system-defined collections.

9.4 Change Scheduled Frequency

This task allows for changing the frequency that the scheduled NR-NSA runs.

Prerequisites

— User has the roles outlined in the Predefined User Setup on page 95 section.

— Scheduled update for NR-NSA is enabled.

Steps

1. Log on, via SSH, to the ENM General Scripting VM that is currently scheduling NR-NSA Custom Topology updates.

   ```
   [litp-ms] $ ssh <predefined-user>@scp-1-scripting
   <predefined-user>@scp-1-scripting's password
   [scp-1-scripting] $
   ```

   Note: Do not log on as another <enmuser> and then run

   ```
   su - <predefined-user>
   ```

   as this bypasses the required Pam authentication.

2. As <predefined-user>, update the cron job for NR-NSA from the <predefined-user> cron:

   ```
   [scp-1-scripting] $ crontab -e
   ```

   Example:

   ```
   0 0 * * * /opt/ericsson/nr-nsa-systems-topology/crontabs/cron.sh 2>/dev/null
   ```

3. Log off ENM General Scripting VM:

   ```
   [scp-1-scripting~]] $ exit
   ```

   Scheduled NR-NSA Topology updates will execute at the new occurrence.

9.5 Manually Execute Script

This task explains how to manually perform the NR-NSA execution.

Prerequisites

User has the roles outlined in the Predefined User Setup on page 95.

Steps

1. Log on, via SSH, to the ENM General Scripting VM that is currently scheduling the NR-NSA Custom Topology:

   See Predefined User Setup on page 95 for more information.

   ```
   [litp-ms] $ ssh <predefined-user>@scp-1-scripting
   ```

```
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run

```
su - <predefined-user>
```

as this bypasses the required Pam authentication.

2. Run the Script:

```
python /opt/ericsson/nr-nsa-systems-topology/main.py
```

Note: Make sure you use the same password as in Step 1 as password validation against the LDAP/PAM does not take place during Step 2.

3. Log off ENM General Scripting VM:

```
[scp-1-scripting~]] $ exit
```

Once the NR-NSA python script has run, the Topology and the system-defined collections will be updated accordingly.

9.6 Disable NR-NSA Scheduling

This task allows for the disabling of the NR-NSA Custom Topology, whether temporary or permanent.

Prerequisites

— User has the roles outlined in the Predefined User Setup on page 95.

— Scheduled update for NR-NSA is enabled.

Steps

1. Log on, via SSH, to the ENM General Scripting VM that is currently scheduling the NR-NSA Custom Topology:

   See Predefined User Setup on page 95 for more information.

```
[litp-ms] $ ssh <predefined-user>@scp-1-scripting
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run

```
su - <predefined-user>
```

as this bypasses the required Pam authentication.

2. As <predefined-user>, remove the cron job for NR-NSA from the <predefined-user> cron:

```
[scp-1-scripting] $ crontab -e
```

Example:

```
0 0 * * * /opt/ericsson/nr-nsa-systems-topology/crontabs/cron.sh 2>/dev/null
```

3. Log off ENM General Scripting VM:

```
[scp-1-scripting~]] $ exit
```

Scheduled NR-NSA Topology updates will stop executing on the ENM General Scripting VM.

9.7 Constraints on NR-NSA

— Any modification made by the user to the NR-NSA topology or any of the system-defined collections will be undone when the script next runs.

— If the user does not contain the necessary roles, or if the Cron job is configured by a user without the necessary roles, the NR-NSA script will fail.

— If the password for the <predefined user> changes, the script will fail.

— Any user added collections that contain the same naming convention as the NR-NSA System Topology or system-defined collections will be deleted if they cause a duplication error.

— The NR-NSA Topology will only exist if there are relationships between EnodeB RadioNodes and GnodeB Baseband RadioNodes.

— The LTE-ERBS collection will only exist if there are eNodeB DU Radio Nodes.

— The LTE-RadioNode collection will only exist if there are eNodeB Baseband Radio Nodes.

— The NR-RadioNode collection will only exist if there are gNodeB Baseband Radio Nodes.

10 MSC Pool Topology

MSC Pool Topology script is a system operated script that provides creation and maintenance of MSC Pool Topology and the system-defined collections for Pools : MSCPool_<pool name>.

MSC Pool Topology script is installed in the ENM scripting cluster during the ENM deployment procedures.

The MSC Pool Topology script is scheduled to run every 6 hours. This is configurable.

The MSC Pool Topology script creates a Custom Topology that depicts the associations between the supported MSC node types and MSC Pool objects.

Note: It is important to note that the MSC Pool Topology maintains itself. If a user adds collections underneath the MSC Pool Topology, they will be removed from the Topology during the next occurrence of the Script execution.

The following tasks are detailed in this section:

— Custom Role Setup.

— Predefined User Setup.

— Enable Scheduled MSC Pool Topology Script.

— Change Scheduled Frequency.

— Manually Execute MSC Pool Topology Script.

— Disable MSC Pool Scheduling.

— Constraints on MSC Pool Topology

10.1 Custom Role Setup

For the MSC Pool Topology and the system-defined collections for Pools to be created, a new custom role is needed

Steps

1. Log on as an ENM administrative privilege user.
2. Launch the ENM Role Management Interface.

3. Create a custom role with the following resources:

   System_created_object

   — Create

   — Delete

Results

A custom user role is created.

10.2 Predefined User Setup

For the MSC Pool Topology and the system-defined collections for Pools to be created, an ENM user with relevant rights must be created.

Prerequisites

You have ENM administrative roles, such as create user and assign rights.

Steps

1. Log on as an ENM administrative privilege user.
2. Launch the ENM User Management Interface.
3. Create an ENM user that is assigned the following rights:
   — Scripting_Operator.
   — Network_Explorer_Administrator.
   — Cmedit_Administrator.
   — Custom role. See Custom Role Setup on page 100 for details.
4. Log off ENM User Management.

Results

You will have created a predefined ENM user, that will be used to run the MSC Pool topology script.

10.3 Enable Scheduled MSC Pool Topology Script

This task allows you to enable the cron job that will run the MSC Pool topology script at regular intervals.

Prerequisites

Predefined ENM user, created with the roles outlined in Predefined User Setup on page 101.

Steps

1. Log on, via SSH, to ENM General Scripting VM:

```
[litp-ms] $ ssh <predefined-user>@scp-1-scripting
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run:

```
su - <predefined-user>
```

as this bypasses the required PAM authentication.

2. As the <predefined-user>, execute the setup.py script, then follow prompts:

```
[scp-1-scripting~] $ cd /opt/ericsson/msc-pool-systems-topology/bin/
[scp-1-scripting~] $ ./setup.py
```

Note: Make sure you use the same password as in Step 1 because password validation against the LDAP/PAM does not take place during Step 2.

3. Log off from the General Scripting VM.

Results

The generated cron job becomes active after 30 minutes.

The MSC Pool Topology script has now been enabled. By default, at every 6 hours, the script will run and perform updates to the MSC Pool Topology and system-defined collections of Pools.

10.4 Change Scheduled Frequency

This task allows for changing the frequency that the scheduled MSC Pool Topology runs.

Prerequisites

— User has the roles outlined in Predefined User Setup on page 101.

— Scheduled update for MSC Pool Topology is enabled.

Steps

1. Log on, via SSH, to the ENM General Scripting VM that is currently scheduling MSC Pool Topology updates.

```
[litp-ms] $ ssh <predefined-user>@scp-1-scripting
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run:

```
su - <predefined-user>
```

as this bypasses the required PAM authentication.

2. As <predefined-user>, update the cron job for MSC Pool Topology from the <predefined-user> cron:

```
[scp-1-scripting] $ crontab -e
```

Example:

```
 0 */6 * * * /opt/ericsson/msc-pool-systems-topology/main.py cron 2>/dev/nul →
l
```

3. Log off ENM General Scripting VM:

```
[scp-1-scripting~]] $ exit
```

Results

Scheduled MSC Pool Topology updates will execute at the new occurrence.

10.5 Manually Execute MSC Pool Topology Script

This task explains how to manually perform the MSC Pool Topology script execution.

Prerequisites

User has the roles outlined in Predefined User Setup on page 101.

Steps

1. Log on, via SSH, to the ENM General Scripting VM that is currently scheduling the MSC Pool Topology:

   See Predefined User Setup on page 101 for more information.

```
[litp-ms] $ ssh <predefined-user>@scp-1-scripting
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run:

```
su - <predefined-user>
```

as this bypasses the required PAM authentication.

2. Run the Script:

```
python /opt/ericsson/msc-pool-systems-topology/main.py
```

Note: Make sure you use the same password as in Step 1 as password validation against the LDAP/PAM does not take place during Step 2.

3. Log off ENM General Scripting VM:

```
[scp-1-scripting~]] $ exit
```

Results

Once the MSC Pool Topology python script has run, the MSC Pool Topology and the system defined collections will be updated accordingly.

10.6 Disable MSC Pool Topology Script Scheduling

This task allows for the disabling of the MSC Pool Topology, whether temporary or permanent.

Prerequisites

— User has the roles outlined in Predefined User Setup on page 101.

— Scheduled update for MSC Pool is enabled.

Steps

1. Log on, via SSH, to the ENM General Scripting VM that is currently scheduling the MSC Pool Topology:

   See Predefined User Setup on page 101 for more information.

```
[litp-ms] $ ssh <predefined-user>@scp-1-scripting
<predefined-user>@scp-1-scripting's password
[scp-1-scripting] $
```

Note: Do not log on as another <enmuser> and then run:

```
su - <predefined-user>
```

as this bypasses the required PAM authentication.

2. As <predefined-user>, remove the cron job for MSC Pool from the <predefined-user> cron:

```
[scp-1-scripting] $ crontab -e
```

Example:

```
0 */6 * * * /opt/ericsson/msc-pool-systems-topology/main.py cron 2>/dev/null
```

3. Log off ENM General Scripting VM:

```
[scp-1-scripting~]] $ exit
```

Results

Scheduled MSC Pool Topology updates will stop executing on the ENM General Scripting VM.

10.7 Constraints on MSC Pool

— Any modification made by the user to the MSC Pool Topology or any of the system-defined collections of Pools will be undone when the script next runs.

— If the user does not contain the necessary roles, or if the cron job is configured by a user without the necessary roles, the MSC Pool Topology script will fail.

— If the password for the <predefined user> changes, the MSC Pool Topology script will fail.

— Any user added collections that contain the same naming convention as the MSC Pool Topology or system-defined collections will be deleted if they cause a duplication error.

— The MSC Pool Topology will exist only if at least one MSC Pool created in ENM.

— The MSC Pool Collection will only exist if there are any MSC Pools created.

11 Node Health Check Administration Tasks

This section contains the routine operation and maintenance tasks related to the administration of the Node Health Check application.

11.1 Configure Node Health Check Housekeeping Parameters

Configure the Node Health Check (NHC) housekeeping parameters to control the amount of storage resources used by NHC. Reports older than the value configured are deleted once a day together with the corresponding jobs from the job list.

When the amount of disk space used by the reports exceeds the maximum value configured, NHC housekeeping deletes the excess starting from the oldest ones.

Note: For details on how to view and modify PIB parameters using service name cmserv, refer to section Configuring PIB Parameters.

Table 32 Parameters to configure Node Health Check

Parameter	Scope	Default Value	Description	Change To
`nhcMaximumFileAgeInDays`	SERVICE	`90`	Maximum retention period for reports.	-
`nhcMaximumDiskSpaceUsageInBytes`	SERVICE	13421772800 (12,5 GB)	Maximum total size of report files	`8000000000`

11.2 Configure Node Health Check Time-Out Parameters for eNodeb Baseband Radio Nodes

The time-out parameters are configurable for ECIM nodes Health Check activity.

Note: For details on how to view and modify PIB parameters using service name shmcoreserv, refer to Configuring PIB Parameters.

Table 33 Node Health Check Activities Time-Out Configuration Parameters for eNodeb Baseband Radio

Parameter	service_identifier	Scope	Default Value	Description	Change To
`ecimNodeHealthCheckActivityTimeOut`	node-health-check-ecim-service	SERVICE	15	Timeout for health check activity of ECIM Node Health Check Report	-

Parameter	service_identifier	Scope	Default Value	Description	Change To
ecimNodeHealthCheckActivityTimeOut	cppinventorysynchservice	SERVICE	15	Maximum total size of report files	8000000000

11.3 Configurable Parameters for Housekeeping of Reports in Node Health Check UI

NHC enables you to create reports to administer the nodes in your network. NHC provides housekeeping policies to ensure that, over time, the completed reports do not cause data storage space issues.

Housekeeping Policies

The housekeeping of the reports runs on a daily basis at 5 AM.

Schedule Time Parameters

Reports Housekeeping policy contains report age criteria and report count criteria.

First, reports cleanup is done based on age criteria. If the count of reports is still more than specified maximum count, then the clean up is done until report count reaches specified maximum count.

1. Report Age Criteria

   When the number of reports exceeds the specified number of days, older reports are deleted.

2. Report Count Criteria

   When the number of reports exceeds the specified count, older reports are deleted.

Daily Schedule Time of Reports housekeeping can be determined from the following two configurable parameters:

— DAILY_SCHEDULE_TIME_IN_HOURS_FOR_JOB_HOUSEKEEPING - It determines the hours of scheduled time of housekeeping.

— DAILY_SCHEDULE_TIME_IN_MINUTES_FOR_JOB_HOUSEKEEPING - It determines the minutes of scheduled time of housekeeping.

For example, if the value of the first parameter is 10 and the value of the second parameter is 30, the daily schedule time of Reports House Keeping is 10h 30min.

11.3.1 Update Default Configured Values

All the parameters listed in Configure Node Health Check Housekeeping Parameters on page 106 are configurable.

Note: For details on how to view and modify PIB parameters using service name shmcoreserv, refer to Configuring PIB Parameters on page 8.

11.3.2 Node Health Check Configuration Parameters for Housekeeping Reports

Parameter	Type	Scope	Default Value	Description
nodeHealthCheckReportCountForHousekeeping	int	SERVICE	100	Maximum count of Node Health Check Reports. If the number of node health check reports exceeds the specified count, older reports are deleted as part of housekeeping.
nodeHealthCheckReportAgeForHousekeeping	int	SERVICE	60 days	If the number of Node Health check reports exceeds the specified number of days, older reports are deleted.

11.4 Configurable Parameters for Profiles Threshold

> NHC enables you to create a Health Check Profile for a node type in your network.

The `profilesThreshold` parameters are configurable for allowed **Node Health Check Profile** count.

Note:
- For details on how to view and modify PIB parameters using service name shmcoreserv, refer to Configuring PIB Parameters on page 8.
- It is not suggested to update this value as it is tuned according to system capabilities.

11.5 Node Health Check Profile Count Configuration Parameters

This section describes the Node Health Check Profile Count Configuration parameters.

Table 34 Node Health Check Profile Count Configuration Parameters

Parameter	Type	Scope	Default Value	Description
profilesThreshold	Integer	SERVICE	100	Maximum allowed health check profiles count configuration.

12 VNF Life Cycle Manager Administration Tasks

> This section contains information for System Administration tasks specific to VNF Life Cycle Manager (VNF-LCM) application in the ENM or Ericsson Orchestrator-Cloud Manager system.

VNF Life Cycle Manager (VNF-LCM) needs to connect with ENM to perform some operations while executing workflows. VNF-LCM does not connect to ENM if VNF-LCM is deployed as a value pack to Ericsson Orchestrator-Cloud Manager. VNF-LCM supports integration with the following products:

- Ericsson Cloud Manager (ECM) as NFV Orchestrator (NFVO), for VNF Orchestration.
- Cloud Execution Environment (CEE) /OpenStack as Virtualized Infrastructure Manager (VIM), where ECM is not present.

Note: If there is a VNF-LCM HA deployment, use VIP of services VM to access the system. Similarly, use internal VIP for DB VM.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.1 Configuring VNF-LCM Access

Configure VNF-LCM for communication with ENM, and ECM or CEE Clouds.

Prerequisites

- Root and cloud-user access to the `vnflaf-services` VM.
- A command console is opened.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.
2. Use the following command to read any ENM parameter listed in VNF-LCM ENM Parameters Table, or any Cloud parameter listed in VNF-LCM Cloud Parameters Table before the update:

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py read --app_ser
ver_address <host name>:8080 --name=<paramName>
```

Example

To read the parameter named `cloudTenantId`:

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py read --app_ser
ver_address localhost:8080 --name=cloudTenantId
```

3. Use the following command to update any ENM parameter listed in VNF-LCM ENM Parameters Table, or any Cloud parameter listed in VNF-LCM Cloud Parameters Table:

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py update --app_s
erver_address <host name>:8080 --name=<paramName> --value=<paramValue> --typ
e=<paramType> --scope=GLOBAL
```

Example

To update the parameter named `cloudTenantId`:

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py update --app_s
erver_address localhost:8080 --name=cloudTenantId --value=Tenant001 --type=S
tring --scope=GLOBAL
```

— RELATED INFORMATION —

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.1.1 VM VNFM ENM Parameters Table

Table 35 Configuration Parameters Specific to ENM

Name	Description	Default Value	Type
`ossType`	Describes the type of OSS.	ENM	String
`ossHostName`	Describes the hostname required to connect to ENM Scripting VM.	-	String
`ossSshPortNumber`	Describes the SSH port number required to connect to ENM Scripting VM.	22	String
`ossFtpPortNumber`	Describes the FTP port number required to connect to ENM Scripting VM.	22	Integer
`ossUserName`	Authorized user of ENM Scripting VM Server for VNF handling.	vnflaf	String

Name	Description	Default Value	Type
ossPassword	Describes the password for above FTP user required to connect to ENM Scripting VM.	-	String
storageLocation	Indicates where the VNF artifacts are located, either remote or local.	local	String
ossSoftwareDir	Indicates the directory path of artifacts required for the VNF.	/home/shared/${ossUserName}/vnflaf	String

The following parameters are supported only in older versions of VNF packages.

Table 36 Deprecated Configuration Parameters

Old Parameter Name	New Parameter Name	Description	Default Value	Type
ossRcMasterHostName	ossHostName	Describes the hostname required to connect to ENM Scripting VM.	-	String
ossRcSshPortNumber	ossSshPortNumber	Describes the SSH port number required to connect to ENM Scripting VM.	22	String
ossRcFtpPortNumber	ossFtpPortNumber	Describes the FTP port number required to connect to ENM Scripting VM.	22	Integer
ossRcFtpUserName	ossUserName	Authorized user of ENM Scripting VM Server for VNF handling.	vnflaf	String
ossRcFtpPassword	ossPassword	Describes the password for above FTP user required to connect to ENM Scripting VM.	-	String
ossRcSshUserName	ossUserName	Authorized user of ENM Scripting VM Server for VNF handling.	vnflaf	String
ossRcSshPassword	ossPassword	Describes the password for above sshuser required to connect to ENM Scripting VM.	-	String

12.1.2 VNF-LCM Cloud Parameters Table

Table 37 Configuration Parameters Specific to Cloud Manager and VIM

Name	Description	Default Value	Type
waitPeriod	Describes time period to wait for responses from ECM.	P0DT0H0M30S	String

Name	Description	Default Value	Type
deleteBsvs	Describes a boolean value which determines whether to delete Bsvs or not.	false	Boolean
cloudROApiVersion	Describes vCloud API schema version.	32.0	String
contrailPort	Describes the port where contrail services are available.	8082	String

The following parameters are supported only in older versions of VNF packages.

Table 38 Deprecated Configuration Parameters

Old Parameter Name	New Parameter Name	Description	Default Value	Type
ecmTenantId	cloudTenantId	Specifies the ECM tenant ID.	VNF-LAF	String
ecmBaseURL	cloudBaseURL	Specifies the base URL to connect the Ericsson Cloud Manager.	https://cloudurl/ecm_service	String
ecmUserName	cloudUserName	The username in ECM to be used by VNF-LCM to create the VNFs.	XID	String
ecmUserPassword	cloudUserPassword	The Password in ECM to be used by VNF-LCM to create the VNFs.	-	String

Note: For more details on how to update any deprecated configuration parameters, see Configuring VNF-LCM Access.

12.1.3 Update Password for ENM

Configure VNF-LCM to update password parameters for ENM.

Prerequisites

— vnflaf-services VM root access.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Update SSH and SFTP passwords of ENM:

```
[root@vnflaf-services ~]# vnflcm oss passwd
```

12.2 Configuring VNF-LCM Cloud Integration

This section describes how to configure VNF-LCM to enable access to ECM or CEE.

Note: For VNF-LCM HA, follow the necessary subsections on all `vnflaf-services` VMs.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.2.1 Verify Certificate Installation

This section describes how to check if the certificate for Cloud is installed in the system.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— A command console is opened.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.
2. Check if certificate is installed correctly:

```
[root@vnflaf-services ~]# <JDKPATH>/bin/keytool -list -v -keystore <JDKPATH> →
 /jre/lib/security/cacerts | grep -i <ecm or cee or openstack or vCloud dire →
ctor>
```

   `<JDKPATH>` is the JDK Installed directory path, for example, `/usr/java/jdk1.8.0_202`.

3. Enter the `keystore` password and press Enter.
4. Display the ECM certificate:

```
root@vnflaf-services ~]#
/usr/java/jdk1.8.0_202/bin/keytool - list -v -keystore          →

/usr/java/jdk1.8.0_202/jre/lib/security/cacerts | grep -i ecm
Enter keystore password: changeit
```

```
Owner:
EMAILADDRESS=ecm-ci@mailman.lmera.ericsson.se,
CN=ieatecm03.athtem.eei.ericsson.se, OU=ECM, O=Ericsson, L=Pi →
scataway,
ST=NJ, C=US Issuer: EMAILADDRESS=ecmci@mailman.lmera.ericsson →
.se,
CN=ieatecm03.athtem.eei.ericsson.se, OU=ECM, O=Ericsson, L=Pi →
scataway,
ST=NJ, C=US
[root@vnflaf-services ~]#
```

If the certificate is not installed, no output is shown. In this case, refer to Install Certificate section to install the certificate.

12.2.2 Check Certificate Expiry

Every Certificates have a validity time period. Check the expiry date of the installed certificate and install it again if the validity period has expired.

The `<JDKPATH>` used in the steps is the JDK installed directory path. For example: `/usr/java/jdk1.8.0_202`.

Prerequisites

- Root and cloud-user access to the `vnflaf-services` VM.
- A command console is opened.
- User knows the alias name used during certificate installation.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.
2. List the certificates associated with the alias used during certificate installation.

```
[root@vnflaf-services ~]# <JDKPATH>/bin/keytool -list -v -keystore <JDKPATH> →
/jre/lib/security/cacerts -alias <CA CERTIFICATE ALIAS>
```

Example

List the certificates associated with alias "ca".

```
[root@vnflaf-services ~]# /usr/java/jdk1.8.0_202/bin/keytool -list -v -keyst →
ore /usr/java/jdk1.8.0_202/jre/lib/security/cacerts -alias ca

Enter keystore password: changeit

Alias name: ca

Creation date: Sep 14, 2016
```

```
Entry type: trustedCertEntry

Owner: EMAILADDRESS=ecm-ci@mailman.lmera.ericsson.se, CN=ieatecm03.athtem.ee →
 → i.ericsson.se, OU=ECM, O=Ericsson, L=Piscataway, ST=NJ, C=US Issuer: EMAI →
LADDRESS=ecm-ci@mailman.lmera.ericsson.se, CN=ieatecm03.athtem.e → ei.ericss →
on.se, OU=ECM, O=Ericsson, L=Piscataway, ST=NJ, C=US Serial number: 88fbd479 →
4dae1799

Valid from: Wed Jul 27 14:57:34 EDT 2016 until: Thu Jul 27 14:57:34 EDT 2017

Certificate fingerprints:

MD5: 89:74:29:76:12:A3:70:18:3E:C7:58:B8:4B:1D:EF:11

SHA1: 22:57:5D:7C:51:CF:20:FB:E9:55:96:5D:F3:BF:B9:29:23:E2:0F:AD SHA256: F4 →
:93:53:F3:83:DC:79:51:1F:27:11:8D:74:44:8A:D3:BB:3B:21:5C → :65:68:D8:10:1B: →
20:FB:62:57:3C:2B:88

Signature algorithm name: SHA1withRSA

Version: 1

[root@vnflaf-services ~]#
```

3. Look for "`Valid from`" line in the output of the command to check the validity period.

   If the validity period of the certificate has expired, refer to section Install Certificate on page 116 to install the certificate.

Results

User knows the validity period of the installed certificate. User can install a new certificate if the validity period of the installed certificate has expired.

12.2.3 Install Certificate

Install the SSL certificate in the `vnflaf-services` VM from Cloud Manager, during installation, or if the certificates have expired. The <JDKPATH> used in the examples is the JDK installed in the directory path, for example: /usr/java/jdk1.8.0_202).

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— A command console is opened.

— User knows the Hostname of Cloud Manager.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, then switch to root user.
2. Use the following command to download the SSL certificate from CEE:

```
openssl s_client -showcerts -connect <CLOUD MANAGER HOSTNAME>:<PORT NUMBER> < /d →
ev/null | openssl x509 -outform DER > /tmp/ssl_cert.cer
```

Where:

— `CLOUD MANAGER HOSTNAME` is name of host where CEE services, for example keystone and Heat, are running.

— `PORT NUMBER` is the port used by SSL(https). Default value is 443.

Note: Check if the downloaded certificate (/tmp/ssl_cert.cer) contains the complete chain of the certificates (from host certificate up to the root CA certificate). If it does not contain the complete chain, you must get the missing link in the certificate chain. You can get this from the respective certificate authority. For private CA certificates, the corresponding Cloud admin needs to be contacted. If it is a public CA certificate, then it can be downloaded from the corresponding web site.

Example

With Successful Output:

```
# openssl s_client -showcerts   -connect  131.160.163.12:443  < /dev/null |  →
openssl x509 -outform PEM > /tmp/ssl_cert.cer
depth=2 C = US, O = "VeriSign, Inc.", OU = VeriSign Trust Network, OU = "(c) →
 2006 VeriSign, Inc. - For authorized use only", CN = VeriSign Class 3 Publi →
c Primary Certification Authority - G5
verify return:1
depth=1 C = US, O = Symantec Corporation, OU = Symantec Trust Network, CN =  →
Symantec Class 3 Secure Server CA - G4
verify return:1
depth=0 C = SE, ST = Stockholm, L = Stockholm, O = Ericsson, OU = IT, CN = < →
hostname>
verify return:1
DONE
```

With Failed Output:

```
openssl s_client -showcerts   -connect  131.160.163.12:443  < /dev/null | op →
enssl x509 -outform PEM > /tmp/ssl_cert.cer
depth=0 C = SE, ST = Stockholm, L = Stockholm, O = Ericsson, OU = IT, CN = i →
eatcee01cic.athtem.eei.ericsson.se
verify error:num=20:unable to get local issuer certificate
verify return:1
depth=0 C = SE, ST = Stockholm, L = Stockholm, O = Ericsson, OU = IT, CN = i →
eatcee01cic.athtem.eei.ericsson.se
verify error:num=27:certificate not trusted
verify return:1
depth=0 C = SE, ST = Stockholm, L = Stockholm, O = Ericsson, OU = IT, CN = i →
eatcee01cic.athtem.eei.ericsson.se
verify error:num=21:unable to verify the first certificate
verify return:1
DONE
```

3. If Cloud Manager is ECM, download the SSL certificate using the following command:

```
[root@vnflaf-services ~]# openssl s_client -showcerts -connect <CloUD MANAGE →
R HOSTNAME>:443 < /dev/null | openssl x509 -outform DER > /tmp/ssl_cert.cer
```

4. Remove the CA certificate alias from the system if it exists.

```
[root@vnflaf-services ~]# <JDKPATH>/bin/keytool -delete -keystore  <JDKPATH> →
/jre/lib/security/cacerts -alias <CA CERTIFICATE ALIAS>
```

Example

```
 [root@vnflaf-services ~]# /usr/java/jdk1.8.0_202/bin/keytool -delete -keyst →
ore /usr/java/jdk1.8.0_202/jre/lib/security/cacerts -alias ca
Enter keystore password: changeit
[root@vnflaf-services ~]#
```

5. Install the certificate into the system.

```
[root@vnflaf-services ~]# <JDKPATH>/bin/keytool -import -alias <CA CERTIFICA →
TE ALIAS> -file /tmp/ssl_cert.cer -keystore <JDKPATH>/jre/lib/security/cacer →
ts -storepass changeit
```

Note: Use different alias where installing more than one certificate.

Example

```
[root@vnflaf-services ~]# /usr/java/jdk1.8.0_202/bin/keytool -import -alias  →
ca -file /tmp/ssl_cert.cer -keystore /usr/java/jdk1.8.0_202/jre/lib/security →
/cacerts -storepass changeit
Owner: EMAILADDRESS=ecm-ci@mailman.lmera.ericsson.se, CN=ieatecm03.athtem.ee →
i.ericsson.se, OU=ECM, O=Ericsson, L=Piscataway, ST=NJ, C=US
Issuer: EMAILADDRESS=ecm-ci@mailman.lmera.ericsson.se, CN=ieatecm03.athtem.e →
ei.ericsson.se, OU=ECM, O=Ericsson, L=Piscataway, ST=NJ, C=US
Serial number: 88fbd4794dae1799
Valid from: Wed Jul 27 14:57:34 EDT 2016 until: Thu Jul 27 14:57:34 EDT 2017
Certificate fingerprints:
         MD5:  89:74:29:76:12:A3:70:18:3E:C7:58:B8:4B:1D:EF:11
         SHA1: 22:57:5D:7C:51:CF:20:FB:E9:55:96:5D:F3:BF:B9:29:23:E2:0F:AD
         SHA256: F4:93:53:F3:83:DC:79:51:1F:27:11:8D:74:44:8A:D3:BB:3B:21:5C →
:65:68:D8:10:1B:20:FB:62:57:3C:2B:88
         Signature algorithm name: SHA1withRSA
         Version: 1
Trust this certificate? [no]:  yes
Certificate was added to keystore
[root@vnflaf-services ~]#
```

6. Update Cloud manager hostname and IP address in `/etc/hosts` and `/vnflcm-ext/current/workflows/multivim_host` files in `VNFLAF-services` VM.

```
echo "<Cloud IP>    <Cloud Hostname>" >> /etc/hosts
echo "<Cloud IP>    <Cloud Hostname>" >> /vnflcm-ext/current/workflows/multi →
vim_host
```

7. Restart jboss for certificate acknowledgment in keystore using following command:

```
[root@vnflaf-services ~]# service jboss restart
```

12.2.4 4G Virtualised RAN Certificate Setup

If you are configuring a 4G Virtualised RAN configuration, the following setup is required to install the relevant certificates on VNF LCM. Alternatively, this can be

done by a script, which will also install the ERICradiovnflcm RPM required for 4G Virtualised RAN workflows.

12.2.4.1 4G Virtualised RAN Certificate Setup (Automated)

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is opened.

— The user knows the Hostname of Cloud Manager.

Steps

1. Download the ENM ISO ERICenm_CXP9027091 from ENM Product Set to a client machine.
2. Extract the artefact ERICradiovnflcm_CXP9034159-<VERSION>.rpm from ENM ISO.
3. Mount the ENM ISO as shown in the following example:

```
$ mkdir /mnt/iso
$ mount -o loop ERICenm_CXP9027091-X.X.X.iso /mnt/iso
```

4. Copy the required rpm located in the directory `/mnt/iso/repos/ENM/services/` on the mounted ISO to the target directory:

```
$ cp /mnt/iso/repos/ENM/services/ERICradiovnflcm_CXP9034159-X.XX.X.rpm <target Directory>
```

5. Update the user <ossUserName> (mentioned in SED file vnflcm-heat-template-static-cinder-env.yaml) with the additional role of PKI_Administrator using ENM User Management application.
6. Extract the ERICradiovnflcm_CXP9034159 rpm on the client machine.
7. On client machine, navigate to directory: `/opt/Ericsson/ERICradiovnflcmworkflows/initial_setup/` which contains 'laf_additional_setup.sh' shell script to perform additional setup.
8. Log on to vnflaf-service VM as cloud-user using external IP assigned or from VM console.
9. Copy the following script to `/tmp/` directory of VNF-LCM service VM.

```
To integrate VNF-LCM with vENM:

$ sudo scp /opt/Ericsson/ERICradiovnflcmworkflows/initial_setup/laf_additional_setup.sh cloud-user@<External_IP_vnflaf-service_VM>:/tmp
```

```
In the case where VNF-LCM need to integrate with Physical ENM :

$ sudo scp /opt/Ericsson/ERICradiovnflcmworkflows/initial_setup/laf_addition →
al_setup_physical.sh cloud-user@<External_IP_vnflaf-service_VM>:/tmp
```

10. Change the permission of script to "755".

```
To integrate VNF-LCM with vENM:

$ sudo chmod 755 /tmp/laf_additional_setup.sh

In the case where VNF-LCM need to integrate with Physical ENM :

$ sudo chmod 755 /tmp/laf_additional_setup_physical.sh
```

11. Confirm that <ossUserName> user can login to <ossHostName> with <ossPassword> password from VNF-LCM service VM. <ossHostName> and <ossUserName> is mentioned in SED file vnflaf-heat-template-static-cinder-env.yaml.

```
Confirm that $ ssh <ossUserName>@<ossHostName>
```

12. **This step is only applicable if you are configuring VNF-LCM for the physical ENM server.** Login to MS VM of Physical ENM server and stop iptables service by executing the following command:

```
$ sudo service iptables stop
```

13. Execute the following script from VNF-LCM service VM:

```
To integrate VNF-LCM with vENM:

$ sudo ./tmp/laf_additional_setup.sh <REPO_NFS_IP>

In the case where VNF-LCM need to integrate with Physical ENM :

$ sudo ./tmp/laf_additional_setup_physical.sh <MS_IP>
```

Note: REPO_NFS_IP can be found in ENM SED file.

14. **This step is only applicable if you are configuring VNF-LCM for the physical ENM server.** Login to MS VM of Physical ENM server and start iptables service by executing the following command:

```
$ sudo service iptables start
```

12.2.4.2 4G Virtualised RAN Certificate Setup (Manual)

Installing certificates on VNF-LCM for 4G Virtualised RAN configuration.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is open.

— The Hostname of Cloud Manager is known.

— PKI_Administrator role assigned to user.

Steps

1. Login to Ericsson Network Manager's ENM CLI application with a user who has PKI_Administrator role.
2. Execute following command to download the CA certificates in PEM format.

```
>> pkiadm certmgmt CACert --exportcert --entityname ENM_OAM_CA --format PEM
>> pkiadm certmgmt CACert --exportcert --entityname ENM_PKI_Root_CA --format
 PEM
```

3. Copy the downloaded files to VNF-LCM service VM's "/home/cloud-user" directory.
4. Login to VNF-LCM service VM with "cloud-user" as the user.
5. Convert the certificate to .crt using the following command:

```
[cloud-user@vnflaf-services ~]$ openssl x509 -in /home/cloud-user/ENM_PKI_Ro
ot_CA.pem -outform der -out /home/cloud-user/ENM_PKI_Root_CA.crt
[cloud-user@vnflaf-services ~]$ openssl x509 -in /home/cloud-user/ENM_OAM_CA
.pem -outform der -out /home/cloud-user/ENM_OAM_CA.crt
```

6. Install the certificates converted in the previous step. Refer to Install Certificate for detailed instructions.

```
[cloud-user@vnflaf-services ~]# sudo /usr/java/default/bin/keytool -import -
alias ENM_PKI_Root_CA -file /home/cloud-user/ENM_PKI_Root_CA.crt -keystore /
usr/java/default/jre/lib/security/cacerts -storepass changeit

[cloud-user@vnflaf-services ~]# sudo /usr/java/default/bin/keytool -import -
alias ENM_OAM_CA -file /home/cloud-user/ENM_OAM_CA.crt -keystore /usr/java/d
efault/jre/lib/security/cacerts -storepass changeit
```

7. Log into ENM with a user with PKI_Administrtaor role and launch ENM CLI application.
8. Create `VnfLcmTrustProfile.xml` file containing the following, then drag-and-drop the file to ENM CLI application

```
<?xml version="1.0" encoding="UTF-8"?>
<Profiles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamesp
aceSchemaLocation="ProfilesSchema.xsd">
    <TrustProfile Name="VNF_LCM_vRAN_Trust_Profile">
        <ProfileValidity>2020-03-01</ProfileValidity>
        <Modifiable>true</Modifiable>
        <TrustCAChain>
            <IsChainRequired>true</IsChainRequired>
            <InternalCA>
                <CertificateAuthority>
                    <Name>NE_OAM_CA</Name>
                </CertificateAuthority>
```

```
            </InternalCA>
        </TrustCAChain>
    </TrustProfile>
</Profiles>
```

9. After the file has copied successfully, execute the following command to create PKI Trust Profile for VNF-LCM.

```
>> pkiadm pfm -c -xf file:VnfLcmTrustProfile.xml
```

10. Create `VnfLcmCertificateProfile.xml` file containing the following, then drag-and-drop the file to ENM CLI application.

```
<?xml version="1.0" encoding="UTF-8"?>
<Profiles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamesp →
aceSchemaLocation="ProfilesSchema.xsd">
    <CertificateProfile Name="VNF_LCM_vRAN_Certificate_Profile">
        <ProfileValidity>2020-05-30</ProfileValidity>
        <ForCAEntity>false</ForCAEntity>
        <Version>V3</Version>
        <SignatureAlgorithm>
            <Name>SHA256withRSA</Name>
        </SignatureAlgorithm>
        <KeyGenerationAlgorithm>
            <Name>RSA</Name>
            <KeySize>2048</KeySize>
        </KeyGenerationAlgorithm>
        <CertificateValidity>P2Y</CertificateValidity>
        <Issuer>
            <CertificateAuthority>
                <Name>ENM_OAM_CA</Name>
            </CertificateAuthority>
        </Issuer>
        <SubjectUniqueIdentifier>false</SubjectUniqueIdentifier>
        <IssuerUniqueIdentifier>false</IssuerUniqueIdentifier>
        <SkewCertificateTime>PT50M</SkewCertificateTime>
        <CertificateExtensions>
            <CertificateExtension xsi:type="AuthorityKeyIdentifier">
                <Critical>false</Critical>
                <AuthorityKeyIdentifierType>SUBJECT_KEY_IDENTIFIER</Authorit →
yKeyIdentifierType>
            </CertificateExtension>
            <CertificateExtension xsi:type="CRLDistributionPoints">
                <Critical>false</Critical>
                <DistributionPoint>
                    <DistributionPointName>
                        <FullName>http://$FQDN_IPV4/pki-cdps?ca_name=$CANAME →
&amp;ca_cert_serialnumber=$CACERTSERIALNUMBER</FullName>
                    </DistributionPointName>
                </DistributionPoint>
                <DistributionPoint>
                    <DistributionPointName>
                        <FullName>http://$FQDN_IPV6/pki-cdps?ca_name=$CANAME →
&amp;ca_cert_serialnumber=$CACERTSERIALNUMBER</FullName>
                    </DistributionPointName>
                </DistributionPoint>
                <DistributionPoint>
                    <DistributionPointName>
                        <FullName>http://$FQDN_DNS/pki-cdps?ca_name=$CANAME& →
amp;ca_cert_serialnumber=$CACERTSERIALNUMBER</FullName>
                    </DistributionPointName>
                </DistributionPoint>
            </CertificateExtension>
            <CertificateExtension xsi:type="BasicConstraints">
                <Critical>true</Critical>
                <IsCA>false</IsCA>
            </CertificateExtension>
            <CertificateExtension xsi:type="SubjectKeyIdentifier">
                <Critical>false</Critical>
                <KeyIdentifier>
                    <Algorithm>
                        <Name>160-BIT_SHA-1</Name>
                    </Algorithm>
```

```
                </KeyIdentifier>
            </CertificateExtension>
            <CertificateExtension xsi:type="KeyUsage">
                <Critical>true</Critical>
                <SupportedKeyUsageType>DIGITAL_SIGNATURE</SupportedKeyUsageT →
ype>
                <SupportedKeyUsageType>KEY_ENCIPHERMENT</SupportedKeyUsageTy →
pe>
                <SupportedKeyUsageType>KEY_AGREEMENT</SupportedKeyUsageType>
            </CertificateExtension>
            <CertificateExtension xsi:type="ExtendedKeyUsage">
                <Critical>false</Critical>
                <SupportedKeyPurposeId>ID_KP_SERVER_AUTH</SupportedKeyPurpos →
eId>
                <SupportedKeyPurposeId>ID_KP_CLIENT_AUTH</SupportedKeyPurpos →
eId>
            </CertificateExtension>
            <CertificateExtension xsi:type="SubjectAltName">
                <Critical>false</Critical>
                <SubjectAltNameField>
                    <Type>DIRECTORY_NAME</Type>
                </SubjectAltNameField>
                <SubjectAltNameField>
                    <Type>IP_ADDRESS</Type>
                </SubjectAltNameField>
            </CertificateExtension>
        </CertificateExtensions>
        <SubjectCapabilities>
            <SubjectField>
                <Type>COMMON_NAME</Type>
            </SubjectField>
            <SubjectField>
                <Type>SURNAME</Type>
            </SubjectField>
            <SubjectField>
                <Type>COUNTRY_NAME</Type>
            </SubjectField>
            <SubjectField>
                <Type>LOCALITY_NAME</Type>
            </SubjectField>
            <SubjectField>
                <Type>STATE</Type>
            </SubjectField>
            <SubjectField>
                <Type>STREET_ADDRESS</Type>
            </SubjectField>
            <SubjectField>
                <Type>ORGANIZATION</Type>
            </SubjectField>
            <SubjectField>
                <Type>ORGANIZATION_UNIT</Type>
            </SubjectField>
            <SubjectField>
                <Type>DN_QUALIFIER</Type>
            </SubjectField>
            <SubjectField>
                <Type>TITLE</Type>
            </SubjectField>
            <SubjectField>
                <Type>GIVEN_NAME</Type>
            </SubjectField>
            <SubjectField>
                <Type>SERIAL_NUMBER</Type>
            </SubjectField>
        </SubjectCapabilities>
    </CertificateProfile>
</Profiles>
```

11. After the file has copied successfully, execute the following command to create PKI Certificate Profile for VNF-LCM.

```
>> pkiadm pfm -c -xf file:VnfLcmCertificateProfile.xml
```

12. Create `VnfLcmEntityProfile.xml` file containing the following, then drag-and-drop the file to ENM CLI application

```
<?xml version="1.0" encoding="UTF-8"?>
<Profiles xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xsi:noNamespa →
ceSchemaLocation="ProfilesSchema.xsd">
    <EntityProfile Name="VNF_LCM_vRAN_Entity_Profile">
        <ProfileValidity>2020-01-01</ProfileValidity>
        <Modifiable>true</Modifiable>
        <Category>
            <Modifiable>false</Modifiable>
            <Name>UNDEFINED</Name>
        </Category>
        <Subject>
            <SubjectField>
                <Type>COMMON_NAME</Type>
                <Value>?</Value>
            </SubjectField>
            <SubjectField>
                <Type>COUNTRY_NAME</Type>
                <Value>SE</Value>
            </SubjectField>
            <SubjectField>
                <Type>ORGANIZATION_UNIT</Type>
                <Value>BUCI_DUAC_NAM</Value>
            </SubjectField>
            <SubjectField>
                <Type>ORGANIZATION</Type>
                <Value>ERICSSON</Value>
            </SubjectField>
        </Subject>
        <SubjectAltName>
            <Critical>false</Critical>
            <SubjectAltNameField>
                <Type>DIRECTORY_NAME</Type>
                <Value xsi:type="SubjectAltNameString" xmlns:xsi="http://www →
.w3.org/2001/XMLSchema-instance">
                    <StringValue>CN=COMUser</StringValue>
                </Value>
            </SubjectAltNameField>
        </SubjectAltName>
        <KeyGenerationAlgorithm>
            <Name>RSA</Name>
            <KeySize>2048</KeySize>
        </KeyGenerationAlgorithm>
        <CertificateProfile Name="VNF_LCM_vRAN_Certificate_Profile" />
        <TrustProfile Name="VNF_LCM_vRAN_Trust_Profile" />
        <KeyUsage>
            <Critical>true</Critical>
            <SupportedKeyUsageType>DIGITAL_SIGNATURE</SupportedKeyUsageType>
            <SupportedKeyUsageType>KEY_ENCIPHERMENT</SupportedKeyUsageType>
            <SupportedKeyUsageType>KEY_AGREEMENT</SupportedKeyUsageType>
        </KeyUsage>
        <ExtendedKeyUsage>
            <Critical>false</Critical>
            <SupportedKeyPurposeId>ID_KP_SERVER_AUTH</SupportedKeyPurposeId>
            <SupportedKeyPurposeId>ID_KP_CLIENT_AUTH</SupportedKeyPurposeId>
        </ExtendedKeyUsage>
    </EntityProfile>
</Profiles>
```

13. After the file has copied successfully, execute the following command to create PKI End-Entity Profile for VNF-LCM.

```
>> pkiadm pfm -c -xf file:VnfLcmEntityProfile.xml
```

14. After successful completion of all the previous steps, create a PKI EndEntity for VNF-LCM. To do this create `VnfLcmEndEntity.xml` file containing the following, then drag-and-drop the file to ENM CLI application.

```
<?xml version="1.0" encoding="UTF-8"?>
<Entities xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"    xsi:noNam →
espaceSchemaLocation="EntitiesSchema.xsd">
    <Entity>
        <PublishCertificatetoTDPS>false</PublishCertificatetoTDPS>
        <EntityProfile Name="VNF_LCM_vRAN_Entity_Profile"/>
        <KeyGenerationAlgorithm>
            <Name>RSA</Name>
            <KeySize>2048</KeySize>
        </KeyGenerationAlgorithm>
        <Category>
            <Modifiable>true</Modifiable>
            <Name>UNDEFINED</Name>
        </Category>
        <EntityInfo>
            <Name>VNF_LCM_vRAN_End_Entity</Name>
            <Subject>
                <SubjectField>
                    <Type>COMMON_NAME</Type>
                    <Value>VNF_LCM_vRAN_End_Entity</Value>
                </SubjectField>
                <SubjectField>
                    <Type>COUNTRY_NAME</Type>
                    <Value>SE</Value>
                </SubjectField>
                <SubjectField>
                    <Type>ORGANIZATION_UNIT</Type>
                    <Value>BUCI_DUAC_NAM</Value>
                </SubjectField>
                <SubjectField>
                    <Type>ORGANIZATION</Type>
                    <Value>ERICSSON</Value>
                </SubjectField>
            </Subject>
            <SubjectAltName>
                <Critical>false</Critical>
                <SubjectAltNameField>
                    <Type>DIRECTORY_NAME</Type>
                    <Value xsi:type="SubjectAltNameString">
                        <StringValue>CN=COMUser</StringValue>
                    </Value>
                </SubjectAltNameField>
            </SubjectAltName>
        </EntityInfo>
    </Entity>
</Entities>
```

15. After the file has copied successfully, execute the following command to create PKI End-Entity for VNF-LCM.

```
>> pkiadm etm -c -xf file:VnfLcmEndEntity.xml
```

16. Execute the following command to download the `VNF_LCM_vRAN_End_Entity.jks` file.

```
>> pkiadm certmgmt EECert --generate -nocsr --entityname VNF_LCM_vRAN_End_En →
tity --format JKS --password changeit
```

17. Copy the downloaded `VNF_LCM_vRAN_End_Entity.jks` file to VNF-LCM service's `/ericsson/vnflcm/data/certs/` directory.

    Note: If the `/ericsson/vnflcm/data/certs/` directory does not exists, create it first.

12.2.5 Set vCloud API version for VCD VIM

> This section describes how to set default API version for VCD VIMs added in VNF-LCM.

Prerequisites

— VNF-LCM is either deployed in VCD or CEE or Openstack.

— Root and cloud-user access to the `vnflaf-services` VM.

— At least one VIM is added in VNF-LCM with vim_type as VCD.

— A command console is opened.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.

   Note: If VNF-LCM deployment is HA, use IP filled in SED for variable <external_ipv4_vip_for_services_vm> or <external_ipv6_vip_for_services_vm>

2. Verify if any VCD VIM is added in VNF-LCM.

   ```
   # vnflcm vim list
   ```

   Example:

   ```
   [root@dep001-vnflaf-services ~]# vnflcm vim list

   +-----------------+----------+----------------+---------------
   +-------------------+---------+

   | VimName | Type | HostName | HostIp | AuthUrl | default |

   +-----------------+----------+------------+---------------
   +-------------------------------------------------+---------+

   | vim_vcd_97  | VCD     | cloud4.site.se      | 134.160.140.85 | https://
   cloud4.site.se:1300/api/version | True |

   +-----------------+----------+----------------------+----------------------
   +-------------------------------------------------+---------+
   ```

3. Verify the default API version set for VCD VIMs.

```
# /ericsson/pib-scripts/etc/config.py read --app_server_addre
ss localhost:8080 --name=cloudROApiVersion
```

Example:

```
[root@dep001-vnflaf-services ~]# /ericsson/pib-scripts/etc/co
nfig.py read --app_server_address localhost:8080 --name=cloud
ROApiVersion
32.0
[root@dep001-vnflaf-services ~]#
```

4. Execute any one of the below commands to set default API version for VCD VIM.

   — Execute below command if added VIM VCD version is 9.7

   ```
   # /ericsson/pib-scripts/etc/config.py update --app_server
   _address localhost:8080 --name=cloudROApiVersion --value=
   32.0  --scope=GLOBAL
   ```

   — Execute below command if added VIM VCD version is 8.10

   ```
   # /ericsson/pib-scripts/etc/config.py update --app_server
   _address localhost:8080 --name=cloudROApiVersion --value=
   20.0  --scope=GLOBAL
   ```

   Tip: This API version is helpful if any workflows do not prompt for the API version of VIM to be used. It means if the workflow does not prompt for API version, the value set in above step will be used by workflow.

12.3 VNF-LCM System Backup and Restore

> This section describes the backup and restore functionalities of the VNF-LCM.

Note:

— For a VNF-LCM HA deployment, the user is blocked from backing up or restoring on a standby VM.

— User is not allowed to modify `/vnflcm-ext/current/workflows/keystore-vnfm.jks` file and the `vnfmKeystorePass` configuration parameter.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.3.1 VNF-LCM System Backup

Every `vnflaf-service` and `vnflaf-db` VM its own cron job. The standard cron job is used to back up the VNF-LCM data at weekly intervals.

The current default cron entry executes the backup script every Sunday, at midnight. The system backup is stored in a file named `vnflcm-<date-time>.tar.gz`, inside the `/vnflcm-ext/backups/` directory. The backup is retained for five weeks.

This is an example of complete path with file name: `/vnflcm-ext/backups/vnflcm-2019-07-04-11-50.tar.gz`.

After extracting the tar.gz file, it will create default directory named `vnflcm-<date-time>`.

Example: `vnflcm-2019-07-04-11-50`.

The default directory has three subdirectories:

db	This subdirectory stores a database backup in compressed files.
config	This subdirectory stores configuration files in compressed format.
jboss_logs	This subdirectory contains the compressed JBoss logs.

Note: The backup files are stored in the `/vnflcm-ext/backups` directory of `vnflaf-services` VM. Hence it is highly recommended to transfer the backup files periodically to external storage. This prevents loss of backup data on disk failure.

Log Files

For each session, a log file is created. This log file is stored in a default location `/vnflcm-ext/current/logs/`. The log contains all important actions taken during the backup and restore process. It can be consulted to make sure that the backup is finished as required.

If there is failure while creating the backup, the user can check the log file for "ERROR" messages. For instance, if the database backup fails, the log file shows the message `[!!ERROR!!] Failed to produce database backup for <<dbname>>`.

12.3.1.1 Change Automatic Backup Schedule Configuration

Change the frequency of the automatic backup cron job schedule.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` and `vnflaf-db` VMs.

— A command console is opened.

— User has working knowledge of crontab.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.
2. Check the frequency of the backup cron job in `vnflaf-services` VM.

```
crontab -l
0 0 * * 0 /opt/ericsson/ERICvnflafsgservice_CXP9032544/scripts/admin/vnflcm_ →
backup.sh 2>> /vnflcm-ext/current/logs//vnflaf_backup.err
```

3. Edit crontab and update the field "`0 0 * * 0`" to the desired frequency.

```
[root@vnflaf-services ~]# crontab -e
```

4. Access the `man` page of crontab to list the valid frequency options available.

```
man crontab
```

Results

The automatic backup schedule configuration is changed.

12.3.1.2 Change the Duration to Retain Backup Files

Perform this task to change the duration to retain backup files.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— A command console is opened.

Steps

1. Log on to `vnflaf-services` VM as `cloud-user` and switch to root user.

2. Edit the /opt/ericsson/ERICvnflafsgservice_CXP9032544/scripts/admin/vnflcm.config file and change the value of WEEKS_TO_KEEP property to a desired value.

```
# How many weeks to keep weekly backups
WEEKS_TO_KEEP=5
```

12.3.1.3 Perform Manual Backup

Create a manual backup of the VNF-LCM system when required.

Prerequisites

— Root and cloud-user access to the vnflaf-services and vnflaf-db VMs.

— A command console is opened.

Note: If additional workflows are installed on the server while backup activity is in process, it can result in inconsistencies.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.

   Note: If VNF-LCM deployment type is HA, execute the same on MASTER VM.

2. Run the backup script to create backup of vnflaf-services system.

```
[root@vnflaf-services ~]#  /opt/ericsson/ERICvnflafsgservice_CXP9032544/scri →
pts/admin/vnflafservices_backup.sh
```

   When the script finishes, it displays the following information:

```
2016_09_09:12:16:04 : BACKUP VNFLAFSERVICES : Services
Backup Completed...directory : /vnflcm-ext/backups/vnflcm-2016-09-09-12-16/
```

3. Write down the directory under which the backup was created, for example, /vnflcm-ext/backups/vnflcm-2016-09-09-12-16/. The same directory is used to create backup for vnflaf-db VM.

4. Log on to vnflaf-db VM as cloud-user and then switch to root user.

   Note: If VNF-LCM deployment type is HA, execute the same on MASTER VM.

5. Run the backup script and provide the directory previously created as an argument.

```
[root@vnflaf-db ~]# /opt/ericsson/ERICvnflafpersistencedb_CXP9032663/sh/vnfl
afdb_backup.sh /vnflcm-ext/backups/vnflcm-2016-09-09-12-16/
```

12.3.1.4 Back Up the VNF-LCM System Using CLI Commands

Take a VNF-LCM system backup using **`vnflcm cli`** command.

Prerequisites

A password-free connection is set up between VNF-LCM VMs.

Steps

Run the following command.

```
# vnflcm system export
```

Note: If VNF-LCM deployment type is HA, execute the same on MASTER VM.

This command takes the system backup into a tar archive file (tar.gz).

Example

```
# vnflcm system export
Running back up in services VM
services VM backup is completed in directory /vnflcm-ext/backups/vnflcm-2018-07-
10-07-11/
login to db for taking back up
running backup command in db
DB backup is completed in directory /vnflcm-ext/backups/vnflcm-2018-07-10-07-11/
Back up is archived to /vnflcm-ext/backups/vnflcm-2018-07-10-07-11.tar.gz
system export completed successfully
```

12.3.2 VNF-LCM System Restore

> This section shows how to restore the VNF-LCM system from the previously created backup.

For information on the backup functionality and how to take backup, refer to VNF-LCM System Backup section.

12.3.2.1 VNF-LCM System Manual Restore

> This section describes the steps to restore the VNF-LCM system that was previously backed up.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` and `vnflaf-db` VMs.

— A command console is opened.

Note: We recommend the following:

— Ensure that there are no active workflow instances executing.

— Take a manual backup of the current system. Doing this enables rollback to current state if necessary.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.

   Note: If VNF-LCM deployment type is HA, execute the same on MASTER VM.

2. Stop the JBoss server.

   ```
   [root@vnflaf-services ~]# service jboss stop
   ```

3. Log on to `vnflaf-db` VM as cloud-user and then switch to root user.

4. Start the restore script and pass the complete backup directory path *<backup directory>*.

   ```
   [root@vnflaf-db ~]# /opt/ericsson/ERICvnflafpersistencedb_CXP9032663/sh/vnfl →
   afdb_restore.sh /vnflcm-ext/backups/vnflcm-2016-09-09-12-16/
   ```

   Executing this command restores the database backup taken on date 2016-09-19, at 12:16 hours.

5. Go back to `vnflaf-services` VM, with root user.

   Note: If VNF-LCM deployment type is HA, execute the same on MASTER VM.

6. Start the restore script and pass the complete backup directory path *<backup directory>*.

```
[root@vnflaf-services ~]# /opt/ericsson/ERICvnflafsgservice_CXP9032544/script →
ts/admin/vnflafservices_restore.sh /vnflcm-ext/backups/vnflcm-2016-09-09-12- →
16/
```

The execution of this example command restores the services backup taken on date 2016-09-19, at 12:16 hours.

Results

After successful execution of the restore script, the System is restored from the backup, and all the required services are operational.

12.3.2.2 System Restore Using CLI Command

Restore VNF-LCM from the backup file.

Prerequisites

— Passwordless connection needs to be setup between VNF-LCM VMs.

— Backup file is copied to `/vnflcm-ext/backups` path in vnflaf-services VM.

Steps

1. Log on to vnflaf-services VM and switch to root user.

   Note: If VNF-LCM deployment type is HA, execute the same on MASTER VM.

2. Execute below command.

   ```
   # sed -i "s/-C \///" /opt/ericsson/ERICvnflafsgservice_CXP9032544/scripts/ad →
   min/vnflafservices_restore.sh
   ```

   The execution of the above command restores the services backup taken on date 2016-09-19, at 12:16 hours.

3. Copy required files related to enm workflows to a directory. The following are the steps to be executed before geo-r import.

   Note: These steps are applicable only if osstype in VNF-LCM SED is vENM.

   a. Create a directory naming enm_data in /tmp/ folder.

      ```
      # mkdir /tmp/enm_data
      ```

   b. Copy directory /vnflcm-ext/enm to /tmp/enm_data.

      ```
      # cp -r /vnflcm-ext/enm /tmp/enm_data
      ```

c. Copy directory /vnflcm-ext/vnf-lcm to /tmp/enm_data.

```
# cp -r /vnflcm-ext/vnf-lcm /tmp/enm_data
```

d. Copy file /vnflcm-ext/current/vnflcm.config to /tmp/enm_data.

```
# cp -r /vnflcm-ext/current/vnflcm.config /tmp/enm_dat
a
```

4. Run any one of the following commands to restore system from backup file.

— Execute the following command to restore system from backup file in a Non-Geo Redundant deployment

```
# vnflcm system import --file <backup_file>
```

Where *<backup_file>* is the backup file(tar.gz) created in Backup Using CLI Command.

Example:

```
[root@vnflaf-services ~]# vnflcm system import --file /vnflcm-ext/backu
ps/vnflcm-2018-07-10-07-11.tar.gz
Login to db for restoring back up
Extracting /vnflcm-ext/backups/vnflcm-2018-07-10-07-11.tar.gz
Running restore command in db
Db restore is completed successfully
Running restore command in services VM
Services restore is completed successfully
system import completed successfully
```

— Execute the following command to restore system from backup file in a Geo Redundant deployment.

```
# vnflcm system import --file <backup_file>  --geo-enabled
```

Where *<backup_file>* is the backup file (tar.gz) created in Backup Using CLI Command, and `--geo-enabled` enables to restore the backup file in a Geo Redundant deployment.

Note: After running the GEO-R import (as part of restore procedure) from primary site to the secondary site, VIM information (tenant and sub-tenant included) will get overridden with primary site vim information (tenant and sub-tenant included). This will cause problems if the VIM information is different across primary and secondary sites. After import is completed, manually add the required VIM information (secondary site VIM details) at secondary site, if required.

Please see:

Add a New VIM on page 148

Add New Domain to a VIM on page 154

5. Copy the required files related to enm workflows from the directory created in step 3 to their respective paths. The following are the steps to be executed after geo-R import is completed.

   Note: These steps are applicable only if osstype in VNF-LCM SED is vENM

   a. Remove below 3 restored files or directories.

      /vnflcm-ext/enm

      /vnflcm-ext/vnf-lcm

      /vnflcm-ext/current/vnflcm.config

      ```
      # rm -rf /vnflcm-ext/enm /vnflcm-ext/vnf-lcm /vnflcm-ext/current/vnflcm.config
      ```

   b. Copy directory /tmp/enm_data/enm to /vnflcm-ext/.

      ```
      # cp -r /tmp/enm_data/enm /vnflcm-ext
      ```

   c. Copy directory /tmp/enm_data/vnf-lcm to /vnflcm-ext/.

      ```
      # cp -r /tmp/enm_data/vnf-lcm /vnflcm-ext
      ```

   d. Copy file /tmp/enm_data/vnflcm.config to /vnflcm-ext/current/.

      ```
      cp -r /tmp/enm_data/vnflcm.config /vnflcm-ext/current/
      ```

   e. Remove directory /tmp/enm_data

      ```
      # rm -rf /tmp/enm_data
      ```

6. If an internal server error occurs while launching any workflow instance after performing import in Geo Redundant deployment, then follow *Unable to Launch a New Workflow Instance after Importing in Geo-Redundant Deployment* in the ENM Configuration Troubleshooting Guide [40]

   Note: This step can be ignored if restore system from backup file is done in Non-Geo Redundant deployment.

12.4 Standard Maintenance Procedures

This section describes the maintenance tasks to ensure that the system is efficiently operating.

Note: For VNF-LCM HA deployment, use the VIP of services VM to access the system. Similarly. use an internal VIP for DB VM.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.4.1 Check the Space Used on the System Disks

Check the space used on the system disks. The file systems cannot reach 100% capacity as this can cause system instability and outage.

Perform this task in a weekly basis.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` and `vnflaf-db` VMs.

— A command console is opened.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.
2. Display a detailed report on the disk space usage in the system.

```
[root@vnflaf-db ~]# df -lh
        Filesystem            Size  Used Avail Use% Mounted on
        /dev/mapper/VolGroup00-LogVol00
        8.1G  1.9G  5.8G  25% /
        tmpfs                 1.9G  4.0K  1.9G   1% /dev/shm
        /dev/vda1             190M   62M  119M  35% /boot
        /dev/vdb              9.8G  114M  9.2G   2% /vnflcm-ext
        [root@vnflaf-db ~]#
```

3. Log on to vnflaf-db VM as cloud-user and then switch to root user.
4. Display a detailed report on the disk space usage in the system.

```
[root@vnflaf-db ~]# df -lh
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                      8.1G  1.9G  5.8G  25% /
tmpfs                 1.9G  4.0K  1.9G   1% /dev/shm
/dev/vda1             190M   62M  119M  35% /boot
/dev/vdb              9.8G  114M  9.2G   2% /vnflcm-ext
[root@vnflaf-db ~]#
```

Results

The system administrator has checked and verified that the file system usage is less than 80%.

12.4.2 Database Administration Tasks

> This section describes the regular system administration tasks with regards to database maintenance and how often they are to be performed.

To ensure efficient database performance, you must monitor the database regularly.

PostgresSQL Directory Structure

The PostgresSQL server is installed in the vnflaf-db VM, in the following location:

path:/var/opt/rh/rh-postgresql94.

This directory stores all PostgresSQLserver configuration files.

The PostgresSQL server is initialized here:

path:/var/opt/rh/rh-postgresql94/lib/pgsql/data,

on the vnflaf-db where Postgres is running. This is the vnlaf-db data directory.

12.4.2.1 Check the Database Disk Space

Check the database disk space. We recommend that you do this manually, every week.

Prerequisites

— Root and cloud-user access to the vnflaf-db VM.

— A command console is opened.

Steps

1. Log on to `vnflaf-db` VM as cloud-user and switch to root user.
2. Display a detailed report on the disk space usage in the system.

```
[root@vnflaf-db ~]# df -lh /vnflcm-ext
Filesystem      Size  Used Avail Use% Mounted on
/dev/vdb        9.8G  158M  9.1G   2% /vnflcm-ext
[root@vnflaf-db ~]#
```

Results

User knows the disk space occupied by the database.

12.4.2.2 Check the Database Size

> This section describes how to check the database size. We recommend that you do this task manually, every week.

Prerequisites

— `Root` and `cloud-user` access to the `vnflaf-db` VM.

— A command console is opened.

Steps

1. Log on to `vnflaf-db` VM as `cloud-user` and switch to `root` user.
2. Switch to `postgres` user.

```
[root@vnflaf-db ~]# su - postgres
```

3. Change to the PostgresSQL bin directory.

```
cd /opt/rh/rh-postgresql94/root/usr/bin
```

4. Get the list of databases.

```
-bash-4.1$ ./psql -l
```

5. Run the query on each database to get its size.

```
postgres=# ./psql SELECT pg_size_pretty(pg_database_size('<db_name>'))
```

Example 5

```
[root@vnflaf-db ~]# su - postgres
-bash-4.1$ cd /opt/rh/postgresql94/root/usr/bin
-bash-4.1$ ./psql -l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileg
es
-----------+----------+----------+-------------+-------------+-----------------
-----
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 sfwkdb    | sfwk     | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
    +
           |          |          |             |             | postgres=CTc/post
gres
 vnflafdb  | vnflaf   | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 wfsdb     | wfs      | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
(6 rows)

-bash-4.1$ ./psql
psql (9.2.7)
Type "help" for help.

postgres=# SELECT pg_size_pretty(pg_database_size('wfsdb'));
 pg_size_pretty
----------------
 12 MB
(1 row)

postgres=#
```

12.4.3 Log Files and Dump Locations on Virtual Machine

> This section gives the locations of the log files inside `vnflaf-services` VM.

It is important for system administrator to reference all the necessary log files and application dumps if they exist.

Jboss Logs

All JBoss logs are stored locally in `/ericsson/3pp/jboss/standalone/log`.

3PP and System Logs

As standard, most 3PP and system logs are available locally in `/var/log`.

command-executor-service

VNF-LCM provides the service to execute the long running script asynchronously.

The `.out` extension file contains output information logged by the script execution, and the `.error` extension file contains any errors related to this task.

The logs relevant to this task are stored under `/var/log/cmd-exec-service` in the `cmd-exec-service.log`file.

	Along with this file, two more logs files are created with `.out` and `.error` extensions.
Dumps	All application memory and core dump files are located in `/ericsson/oss/dumps/`.

Contact local Ericsson support if the result is not as expected.

12.5 Housekeeping VNF-LCM Data

VNF-LCM persists data within its DB. This data grows quickly, and has an impact on the performance of the framework. As data grows, UI performance is affected when there is a large amount of instance-related data. Purging of logs is an approach to help control the data growth.

Note: For a VNF-LCM HA deployment, use VIP of services VM to access the system. Similarly, use an internal VIP for DB VM.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.5.1 Purging Workflow Logs

This section describes the purging of progress events of type "log", from the workflow service database, that are older than a specific duration.

The standard cron job is used to purge the logs. Cron job is scheduled as part of vnflaf-db VM. Data older than the specified purge duration is removed by the cron job at regular intervals.

The default value for PROGRESSEVENT_LOG_PURGE_DURATION is 7 days. This is configurable as per Change the Duration to Purge Logs on page 140.

By default, the cron job on vnflaf-db VM runs daily at 2 AM to purge the workflow logs. This is configurable as per Change Automatic Purge Schedule Configuration on page 141.

To retain data, backup to an external system prior to purging, as per VNF-LCM System Backup and Restore on page 127.

Note: Switch to root user when executing the scripts to have the correct permissions.

12.5.1.1 Change the Duration to Purge Logs

Perform this task to change the duration after which to purge the logs.

Prerequisites

— Root and `cloud-user` access to the `vnflaf-db` VMs.

— A command console is open.

Steps

1. Log in to `vnflaf-db` VM as `cloud-user`, then switch to `root` user.
2. Edit the `/opt/ericsson/ERICvnflafsgdb_CXP9032545/conf/vnflcm_config.conf` file.
3. Change the value of `PROGRESSEVENT_LOG_PURGE_DURATION` property to the desired value.

   Example

```
      #Specifies the duration to delete the progress events of type log wher →
e duration
      can accept second, minute, hour, day, week, month, quarter, year, deca →
de, century,
      millennium e.g. 7 DAYS, 1 MINUTE, 23 HOURS, 1 DAY, 7 WEEKS etc.

      PROGRESSEVENT_LOG_PURGE_DURATION='7 DAYS'
```

12.5.1.2 Change Automatic Purge Schedule Configuration

Procedure to change the frequency of cron job.

Prerequisites

— Root and cloud-user access to the vnflaf-db VMs.

— A command console is open.

— User has working knowledge of crontab.

Steps

1. Login to vnflaf-db VM as cloud-user and then switch to root user.
2. Check the frequency of the purging cron job in vnflaf-db VM by executing crontab -l.

```
      [root@vnflaf-services ~]# crontab -l
      0 2 * * * /opt/ericsson/ERICvnflafsgdb_CXP9032545/bin/purge_cronjobs/w →
fsdb/purge_wfsdb_progressevents_log.sh
```

3. Enter edit mode by executing crontab -e.
4. Update the field 0 2 * * * based on the frequency of purging the progress events of type log.
5. Access the man page of crontab to list the valid frequency options available.

```
man crontab
```

Results

The purging cron job frequency is changed.

12.5.1.3 Enable or Disable the Automatic Purging Schedule

Procedure to enable or disable the scheduled cron job.

Prerequisites

— Root and cloud-user access to the vnflaf-db VMs.

— A command console is open.

— User has working knowledge of crontab.

Steps

1. Login to vnflaf-db VM as cloud-user, then switch to root user.
2. Check the cron job entry in vnflaf-db VM by executing **crontab -l**.

```
[root@vnflaf-services ~]# crontab -l
0 2 * * * /opt/ericsson/ERICvnflafsgdb_CXP9032545/bin/purge_cronjobs/w →
fsdb/purge_wfsdb_progressevents_log.sh
```

3. Enter edit mode by executing **crontab -e**.

If you want to:	Then:
Disable the cron job:	Insert **#** at position 1 in the string: `#0 2 * * * /opt/ericsson/ERICvnflafsgdb_CXP9032545/bin/purge_cronjobs/wfsdb/purge_wfsdb_progressevents_log.sh`

If you want to:	Then:
	Note: Disabling the cron job will result in growth in the amount of data stored. This can cause performance issues. Ensure that manual purging is performed regularly. Refer to Manual Purge of Logs for details.
Enable the cron job:	Remove **#** from position 1 in the string: `0 2 * * * /opt/ericsson/ERICvnflafsgdb_CXP9032545 /bin/purge_cronjobs/wfsdb/purge_wfsdb_progresseve nts_log.sh`

12.5.1.4 Manual Purge of Logs

Apart from the automatic purging of logs via cron job, it is also possible to remove the progress events of type "log", as needed, by executing the purge_wfsdb_progressevents_log script.

Prerequisites

— Root and cloud-user access to the vnflaf-db VMs.

— A command console is open.

Steps

1. Login to vnflaf-db VM as cloud-user, then switch to root user.
2. Change directory to **`/opt/ericsson/ERICvnflafsgdb_CXP9032545/bin`**.

```
cd /opt/ericsson/ERICvnflafsgdb_CXP9032545/bin
```

3. Run the purging script `/purge_cronjobs/wfsdb/purge_wfsdb_progressevents_log.sh`.

```
sh /purge_cronjobs/wfsdb/purge_wfsdb_progressevents_log.sh
```

12.6 Monitoring Application Failure

Monitoring application failure in VNF-LCM includes monitoring of JBOSS, HTTPD, and PostgreSQL failure. Application failure is monitored in the following ways:

— By an external agent

— By a self monitoring script that the user can configure.

This option is available if no external agent is available.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.6.1 Monitor Application Failures with External Agent

> Monitor application failures for JBOSS and HTTPD using an external agent. We recommend that you configure the check for every two minutes.

Note: For a VNF-LCM HA deployment, use the VIP of the VM for the service to access the system. Similarly, use the internal VIP for DB VM.

Prerequisites

User has an external agent or an orchestrator to monitor the services.

Steps

1. Check if the `vmmonitord` service is running in both vnflcm-services and vnflcm-db VMs.

    a. Log on to `vnflaf-services` VM as cloud-user and switch to root user.

    b. Check if `vmmonitord` is running in the VM.

    ```
    #service vmmonitord status
    ```

    Example

    ```
    [root@sitedep001-vnflafservices-0 cloud-user]#service vmmonitord status
    ```

    Note: Repeat the step for `vnflaf-db` also.

2. Perform an HTTP GET call on the http://vnflaf-services:12987/ URL from the external agent, to get the status of applications from both `vnflaf-services` and `vnflaf-db` VM.

    Possible response codes:

    Status code `200` = JBOSS, HTTPD, and PostgresSQL services are running.

Status code 501 = JBOSS and HTTP services are running, but PostgresSQL is not running.

Status code 502 = PostgresSQL is running, but either JBOSS or HTTP services are not running.

Status code 503 = All services (JBOSS, HTTPD, and PostgresSQL) are not running.

Example

Response 503

```
[root@vnflaf-db /]# curl -v http://vnflaf-services:12987/
* About to connect() to vnflaf-services port 12987 (#0)
* Trying 172.16.100.3... connected
* Connected to vnflaf-services (172.16.100.3) port 12987 (#0)
> GET / HTTP/1.1
>
 User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7
NSS/3.16.2.3 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: vnflaf-services:12987
> Accept: */*
> * HTTP 1.0, assume close after body
< HTTP/1.0 503 Service Unavailable
< Server: SimpleHTTP/0.6 Python/2.6.6
< Date: Tue, 11 Jul 2017 20:47:59 GMT
< Content-Type: text/html < Content-Length: 0
< * Closing connection #0
```

3. Take the following Corrective action to deal with response codes:

    a. Complete the following corrective action for Openstack/CEE: Update VNF-LCM stack:

        i. Find and delete VNF-LCM instances using the following commands:

```
openstack stack resource list -n3 <stack name> | grep -w OS::No →
va::Server | awk -F "|" '{print $3}'
openstack server delete <Services server id>
openstack server delete <Db Server id>
```

        Example:

```
# openstack stack resource list -n3 site0001_VNFLCM | grep -w O →
S::Nova::Server | awk -F "|" '{print $3}'
 5882e8b4-b421-4198-9ea5-c74d37af3cbb
 2403e731-c8cf-4dd8-ab21-74b8a52cf89d
# openstack server delete 5882e8b4-b421-4198-9ea5-c74d37af3cbb
# openstack server delete 2403e731-c8cf-4dd8-ab21-74b8a52cf89d
```

        ii. Find and delete VNF-LCM inner stacks:

```
openstack stack resource list -n1 <stack name> | grep -w 0 | aw →
k -F "|" '{print $7}'
openstack stack delete <inner services stack name>
openstack stack delete <inner db stack name>
```

        Example:

```
# openstack stack resource list -n1 site0001_VNFLCM | grep -w 0 →
 | awk -F "|" '{print $7}'
 site0001_VNFLCM-vnflaf-services-ih4fcz7aeibg
 site0001_VNFLCM-vnflaf-db-wvccen47ueao
# openstack stack delete site0001_VNFLCM-vnflaf-services-ih4fcz →
7aeibg
Are you sure you want to delete this stack(s) [y/N]? y
# openstack stack delete site0001_VNFLCM-vnflaf-db-wvccen47ueao
Are you sure you want to delete this stack(s) [y/N]? y
```

iii. Mark inner stack resource unhealthy

```
openstack stack resource list -n1 <stack name> | grep -w OS::He →
at::ResourceGroup | awk -F "|" '{print $2}'
os stack resource mark unhealthy <stack name> <Inner stack reso →
urce name>
```

Example:

```
# openstack stack resource list -n1 site0001_VNFLCM | grep -w O →
S::Heat::ResourceGroup | awk -F "|" '{print $2}'
 vnflaf-services
 vnflaf-db
# openstack stack resource mark unhealthy site0001_VNFLCM vnfla →
f-services
# openstack stack resource mark unhealthy site0001_VNFLCM vnfla →
f-db
```

iv. Update VNF-LCM stack:

```
openstack stack update <stack name> --existing
```

Example:

```
# openstack stack update site0001_VNFLCM --existing
```

b. VCD: If the user needs to be on the same software version, then create a VAPP. Refer to "VNF-LCM Installation Instructions" for more details on how to create VAPP.

12.6.2 Configure Self-monitoring Service

> Self-monitoring feature is activated by the vnflaf-service and vnflaf-db scripts that execute the health check scripts. If the health check finds a service stopped, it retries the check for three times, then restart the corresponding service. It's recommended to configure the check for every two minutes

Prerequisites

— User doesn't have an external agent or an orchestrator.

— cron service is enabled in the VM's images.

Steps

1. Add cron entries to `/var/spool/cron/root`

   vnflaf-service:

   ```
   */2 * * * * /opt/ericsson/ERICvnflafsgservice_CXP9032544/bin/jboss_and_httpd →
   _monitor_scripts.sh >> /tmp/vnflaf_service_monitor.log
   ```

   vnflaf-db:

   ```
   */2 * * * * /opt/ericsson/ERICvnflafsgdb_CXP9032545/bin/db_monitor.sh >> /tm →
   p/vnflaf_db_monitor.log
   ```

12.7 Virtual Infrastructure Manager

> The VNF-LCM Admin CLI is used to configure the connection information of Virtualized Infrastructure Manager (VIM) in VNF-LCM framework. This information is then used by the workflows to connect with VIM and perform operations.

Note: If an VNF-LCM HA deployment, use VIP of services VM to access the system. Similarly, use the internal VIP for DB VM. The user is blocked from performing any activity on standby VM.

Add a New VIM

Add VIM connection information to VNF-LCM framework, which is used by workflows to perform M@M logon and perform operations on VIM.

Observe the following while adding a VIM:

— Add one default VIM to ensure successful execution of workflows in non-interactive mode.

— A default VIM should have one default domain and one default project.

— CEE, OpenStack, and VCD are the supported Cloud types.

— Clusters is optional parameter. If clusters is present in template then `stackId` and `clusterName` are mandatory parameters.

Add a New Domain to an Existing VIM

Add a new domain to connection with VIM. A `domain` can contain multiple `projects`.

Add Cluster

Clusters is optional parameter. If clusters is present in template then `stackId` and `clusterName` are mandatory parameters. Based on authentication type (v2 or v3) vnflcm add cluster details `clusterinv2` or `clusterinv3` table. Later these details require worker node creation.

Delete VIM from VNF-LCM Framework

Existing VIM can be deleted if the connection details are not required anymore.

List the Configured VIMs in the Framework

Lists all configured VIMs in a framework.

We recommend that you list all VIMs before adding a new one, and ensure that one default VIM is present.

Only one default VIM can be present in VNF-LCM framework.

This operation does not display `domain` details of the VIM.

List Project Details of a VIM

Display Project information, which is configured for VIMs with v2 Keystone/ Identity service.

Display Domain and Project information, which is configured for VIMs with v3 Keystone/Identity service.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.7.1 Add a New VIM

> Add VIM connection information to VNF-LCM framework, which is used by workflows to perform M@M logon and perform operations on VIM.

Ad VIM connection information to VNF-LCM framework, which is used by workflows to perform M@M logon and perform operations on VIM.

It is possible to add a second VIM to VNF-LCM, however only one default VIM is allowed. Refer to section Add a second VIM to VNF-LCM to learn more about adding more VIMs.

In the case of CEE and OpenStack, if identity and orchestration host services are running on different hosts, then a VIM with only identity details must be added. Admin cli tool will install the corresponding orchestration certificate. However, if orchestration certificate fails to install, VIM addition will be successful with the identity certificate installed and user needs to install orchestration certificate manually as per Install Certificate on page 116.

Note: Do not add vim with orchestration details, `vim_url`, as orchestration url.

If certificates are modified or expired in openstack or CEE then the user should manually install the certificate from Install Certificate on page 116.

Prerequisites

- VIM user with admin privileges in the VNF-LCM application
- Basic understanding of JSON format.
- Back up the `/ericsson/vnflcm/data/template_vim.json` file before execute this procedure.
- Requires DNS server configured in VNF-LCM.

Steps

1. Edit the backup json file to update its properties.

   ```
   [root@vnflaf-servcies tmp] vi template_vim.json
   ```

   Provide complete file path in case the file is not present in the same directory

   Refer to template_vim.json File section to learn more about this file.

2. Add a new VIM to VNF-LCM framework:

   ```
   [root@vnflaf-services tmp] vnflcm vim add --file=/tmp/template_vim.json
   ```

   Result:

   ```
   Validations passed
   Configuring certificates...
   depth=3 C = US, O = "VeriSign, Inc.", OU = Class 3 Public Primary Certificat →
   ion Authority
   verify return:1
   depth=2 C = US, O = "VeriSign, Inc.", OU = VeriSign Trust Network, OU = "(c) →
    2006 VeriSign, Inc. - For authorized use only", CN = VeriSign Class 3 Publi →
   c Primary Certification Authority - G5
   verify return:1
   depth=1 C = US, O = Symantec Corporation, OU = Symantec Trust Network, CN =  →
   Symantec Class 3 Secure Server CA - G4
   verify return:1
   depth=0 C = SE, ST = Stockholm, L = Stockholm, O = Ericsson, OU = IT, CN = i →
   eatatlascloud2.athtem.eei.ericsson.se
   ```

```
verify return:1
Jboss restarting...
DONE
Certificate was added to keystore
VIM           | Result
--------------+--------------------------------
vim1         | VIM addition successful
```

12.7.1.1 template_vim.json File

Description of the `template_vim.json` file, which is used to add a new VIM to VNF-LCM framework. The file is stored at `/ericsson/vnflcm/data` folder.

```
{
"vims":[{
    "name":"",
    "type":"CEE",
    "defaultVim":"True",
    "hostIpAddress":"",
    "hostName":"",
    "authUrl":"",
    "domain":[{
        "userDomain":"",
        "name":"",
        "id":"",
        "defaultDomain":"True",
        "clusters":[{
            "stackId":"",
            "clusterName":""
        }],
        "project":[{
            "name":"",
            "id":"",
            "username":"",
            "password":"",
            "defaultProject":"True",
            "clusters":[{
                "stackId":"",
                "clusterName":""
            }]
        },
        {
            "name":"",
            "id":"",
            "username":"",
            "password":"",
            "defaultProject":""
        }]
    }]
 }]
}
```

Table 39 template_vim.json File Properties

Attribute	Required	Description
`name`	Y	A user provided unique name to identify the VIM. Workflows developed using SDK of VNF-LCM displays this name to end user.
`type`	Y	Depending on type of VIM, use one of these values: CEE, OpenStack, VCD.

Attribute	Required	Description
defaultVim	Y	Set it to True if the VIM is the default VIM when workflows execute in non-interactive mode. — It is mandatory to have one VIM set as default for successful execution of workflows in non-interactive mode. — Only one VIM can be set as default. — If this property is not set, then it is considered as User Domain configured at Cloud platform. Applicable for OpenStack/CEE with v3 Keystone/ Identity service.False.
hostIpAddress	Y	IP address of VIM.
hostName	Y	Hostname of the VIM. This entry is added in host file of VNF-LCM framework during addition of VIM.
authUrl	Y	REST URL corresponding to Cloud authentication service. Sample URLs for reference. User Domain configured at Cloud platform. Applicable for CEE & OpenStack: https://ieatatlascloud2.athtem.eei.ericsson.se:443/v2 VCD: https://atvplvcd22-v6.athtem.eei.ericsson.se/api
domain	Y	Array containing domain information for VIM. If multiple domains are used for connection, then multiple domain information is provided in array. Workflows provide an option to select one domain among multiple in case of interactive mode. In non-interactive mode, default domain is used for execution. For domains properties, see 'VIM domain Array properties' table.
project	Y	Multiple projects can be added. One default project is mandatory for execution of workflow in non-interactive mode. For projects properties, see *VIM projects Array properties* table.

Table 40 VIM domain Array Properties

Attribute	Required	Description
userDomain	N	User Domain configured at Cloud platform. Applicable for OpenStack/CEE with v3 Keystone/Identity service
name	N	Mandatory for OpenStack/CEE with v3 Keystone/Identity service. This parameter is provided with domain Name configured in a Cloud platform.
id	N	Mandatory for OpenStack/CEE with v3 Keystone/Identity service. This parameter is provided with domain Id configured in a Cloud platform.
defaultDomain	Y	Set this to True if this tenant is the default Tenant when workflows execute in non-interactive mode.
clusters	N	Optional, Array contains cluster information. If v2 vim authentication, then vnflcm insert cluster information in clusterinv2 table. Later vnflcm use these details for worker node creation.

Table 41 VIM project Array Properties

Attribute	Required	Description
name	Y	Project name configured at Cloud platform.
id	Y	Project Id configured in a cloud platform. This is used for connection in OpenStack or CEE flows.
username	Y	User configured at Cloud platform for subtenant.
password	Y	Password of the user.

Attribute	Required	Description
`defaultProject`	Y	Set this to `True` if this project the default project when workflows execute in non-interactive mode. — If `projects` are added, then one among them must be default. — Only one project can be set as default. — If this property is not set, then it is considered as `False`.
`clusters`	N	Optional, Array contains cluster information. If v3 vim authentication, then vnflcm insert cluster information in clusterinv3 table. Later vnflcm use these details for worker node creation.

Table 42 clusters Array Properties

Attribute	Required	Description
`stackId`	Y	Id of stack, where worker node should be create.
`clusterName`	Y	Same cluster name must be configured in EVNFM

12.7.2 Add a Second VIM

> Add a second VIM to replace the existing default VIM in VNF-LCM framework.

Note: If the VIM default property is set to `false`, see Add a New VIM to VNF-LCM Framework before running this procedure.

Prerequisites

— A default VIM must be present in the VM.

— Back up the `/ericsson/vnflcm/data/template_second_vim.json` file before running this procedure.

Steps

1. Fetch the list of the VIM's:

```
[root@vnflaf-servcies tmp] vnflcm vim list
```

2. Delete the existing default VIM:

```
[root@vnflaf-servcies tmp]vnflcm vim delete --name=vim1
```

3. Edit the backup json file:

```
[root@vnflaf-servcies tmp]vi template_second_vim.json
```

Provide the complete file path if the file is not present in the same directory.

See template_second_vim.json File section to learn more about this file.

4. Set the `default` parameter as `true`.

```
"defaultVim": "True"
```

5. Add the second VIM to VNF-LCM framework:

```
[root@vnflaf-services tmp] vnflcm vim add --file=/tmp/template_second_vim.js
on
```

Result:

```
Validations passed
Configuring certificates...
depth=3 C = US, O = "VeriSign, Inc.", OU = Class 3 Public Primary Certificat
ion Authority
verify return:1
depth=2 C = US, O = "VeriSign, Inc.", OU = VeriSign Trust Network, OU = "(c
)2006 VeriSign, Inc. - For authorized use only", CN = VeriSign Class 3 Publi
c Primary Certification Authority - G5
verify return:1
depth=1 C = US, O = Symantec Corporation, OU = Symantec Trust Network, CN =S
ymantec Class 3 Secure Server CA - G4
verify return:1
depth=0 C = SE, ST = Stockholm, L = Stockholm, O = Ericsson, OU = IT, CN = i
eatatlascloud2.athtem.eei.ericsson.se
verify return:1
Jboss restarting...
DONE
Certificate was added to keystore
VIM | Result
-------------+--------------------------------
vim1 | VIM addition successful
vim2 | VIM addition successful
```

12.7.2.1 template_second_vim.json File

> Description of the `template_second_vim.json` file, which is used to add a second VIM to VNF-LCM framework.

```
   {
"vims": [{
"name": "vim1",
"type": "CEE",
"defaultVim": "False",
"hostIpAddress": "10.1.1.1",
"hostName": "ieatatlascloud2.athtem.eei.ericsson.se",
"authUrl": "https://ieatatlascloud2.athtem.eei.ericsson.se:443/v2.0",
"tenants": [{
"name": "nfv",
"id": "432525",
"username": "test1",
"password": "test1",
"defaultTenant": "True",
"clusters": [{
"stackId": "",
```

```
"clusterName": ""
}]
}]
}, {
"name": "vim2",
"type": "OPENSTACK",
"defaultVim": "True",
"hostIpAddress": "10.1.1.2",
"hostName": "cloud12a.athtem.eei.ericsson.se",
"authUrl": " https://cloud12a.athtem.eei.ericsson.se/v3",
"tenants": [{
"userDomain": "",
"name": "nfv",
"id": "453",
"defaultTenant": "False",
"subTenants": [{
"name": "sub1",
"username": "test",
"password": "test",
"defaultSubTenant": "True"
},
{
"name": "sub2",
"username": "test",
"password": "test",
"defaultSubTenant": "False"
}
]
}]
}]
}
```

12.7.3 Add New Domain to a VIM

> Add a new domain to the existing VIM connection. In the case of OpenStack/CEE with v2.0 Keystone/Identity service, the domain parameters such as "userDomain", "name", "id" are left empty.

Prerequisites

— A VIM is already added.

— Back up the `template_domain.json` file before running this procedure.

Steps

1. Fetch the VIM name as follows:

   ```
   [root@vnflaf-services tmp]vnflcm vim list
   ```

2. Edit the properties of `template_domain.json` file.

   ```
   [root@vnflaf-servcies data] vi template_domain.json
   ```

   Provide the complete file path if the file is not present in the same directory.

   See template_domain.json File to learn more about this file.

3. Add `tenant` details to the file as follows:

```
[root@vnflaf-services data]vnflcm vim add-domain  --name=VIM-1 --file=templa →
te_domain.json
```

* The below command is deprecated.

```
[root@vnflaf-services data] vnflcm vim add-tenant --name=VIM- →
1 --file= template_tenant.json
```

Result:

```
Validations passed
domain   | Result
---------+-----------------------------------
domain-1 | Domain  addition successful
```

12.7.3.1 template_domain.json

> Description of the `template_domain.json` file, which is used to add a new tenant to a VIM. The file is stored at `/ericsson/vnflcm/data` folder.

```
{
"domain":[{
"userDomain":"",
"name":"",
"id":"565",
"defaultDomain":"false",
"clusters":[{
"stackId":"",
"clusterName":""
}],
"project":[{
"name":"",
"id":"566",
"username":"admin21",
"password":"admin123",
"defaultProject":"false",
"clusters":[{
"stackId":"",
"clusterName":""
}]
}]
}]
}
```

12.7.4 Delete a VIM

Existing VIM information can be deleted if the connection details are not needed anymore. This deletes the connection information from VNF-LCM database along with the certificates for that VIM.

Prerequisites

— Already added VIM.

Steps

1. Fetch the VIM name:

```
[root@vnflaf-services common]# vnflcm vim list
```

Example

```
VimName          | Type      | HostName                                  | HostI →
p           | AuthUrl                          | default
-----------------+-----------+-------------------------------------------+------ →
-----------+----------------------------------+--------
VIM-1            | CEE       | ieatcee01cic.athtem.eei.ericsson.se       | 131.1 →
60.163.9    | http://141.137.212.4:5000/v2.0   | True
vim2             | OPENSTACK | cloud12a.athtem.eei.ericsson.se           | 131.1 →
60.163.10   | https://cloud12a.athtem.eei.ericsson.se/v3   | False
```

2. Delete the relevant VIM:

```
[root@vnflaf-services demo]# vnflcm vim delete --name=VIM-1
```

Result: VIM deleted successfully.

12.7.5 List Domain Details

> Display existing domain and project details configured for a VIM in case of OpenStack/CEE with v3 Keystone/Identity service.

Note: Display existing project details configured for a VIM in case of OpenStack/CEE with v2 Keystone/Identity service.

Prerequisites

— VIM user with admin privileges.

Steps

1. Fetch the VIM name.

```
[root@vnflaf-services common]# vnflcm vim list
```

Example

```
Name    | Type   | HostName                                  | HostIp           |   →
AuthUrl                          | default
--------+--------+-------------------------------------------+------------------+- →
```

```
--------------------------------+--------
vim1   | CEE   | ieatcee01cic.athtem.eei.ericsson.se    | 131.160.163.9   |
http://141.137.212.4:5000/v2.0  | True
vim2   | OPENSTACK | cloud12a.athtem.eei.ericsson.se    | 131.160.163.10  |
https://cloud12a.athtem.eei.ericsson.se/v3  | False
```

2. List domain details for the relevant VIM.

```
[root@vnflaf-services data]vnflcm vim list-domain --name=vim1
```

*The following command is deprecated.

```
[root@vnflaf-services data] vnflcm vim list-tenant --name=vim
1
```

Example

```
Name | Type | HostName                                    | HostIp          |
AuthUrl                          | default
-----+------------+-----------+-------------------------+-----------------+-
----------------------------------+--------
vim1 | CEE        | ieatcee01cic.athtem.eei.ericsson.se | 131.160.163.9 | ht
tp://141.137.212.4:5000/v2.0 | True
```

12.7.6 List the Configured VIMs

> List all configured VIMs in a framework. We recommend that you list all VIMs before adding a new one, and ensure that one default VIM is present.

Prerequisites

VIM user with admin privileges

Steps

1. List all configured VIMs as follows:

```
[root@vnflaf-services common]# vnflcm vim list
```

Result:

```
Name   | Type  | HostName                                    | HostIp          |
AuthUrl                          | default
-------+-------+---------------------------------------------+-----------------+-
--------------------------------+--------
vim1   | CEE   | ieatcee01cic.athtem.eei.ericsson.se    | 131.160.163.9   |
http://141.137.212.4:5000/v2.0  | True
vim2   | OPENSTACK | cloud12a.athtem.eei.ericsson.se    | 131.160.163.10  |
https://cloud12a.athtem.eei.ericsson.se/v3  | False
vim3   | VCD   | ieatvcd01cic.athtem.eei.ericsson.se    | 131.160.163.11  |
http://141.137.212.6/service    | False
```

12.7.7 Update VIM

Update existing VIM details that are already configured in VNF-LCM. This CLI can be used to update already existing keystone v2.0 to v2.0, v2.0 to v3 or v3 to v3 based keystone authentication.

Note: This CLI does not support downgrading of keystone version from v3 to v2.0.

Prerequisites

— A VIM is already added.

— Back up the `/ericsson/vnflcm/data/template_update_vim.json` before running this procedure.

Steps

1. Fetch the list of already added VIMs:

```
[root@vnflaf-services-0]# vnflcm vim list
```

2. Edit the backup json file:

```
[root@vnflaf-services-0 data]# vi template_update_vim.json
```

Provide the complete file path if the file is not present in the same directory.

See template_update_vim.json to learn more about this file.

3. Update VIM details for VIM already added in VNF-LCM framework:

```
[root@vnflaf-services-0 data]# vnflcm vim update --file template_update_vim. →
json
```

Result:

```
Validations passed
+--------+---------------------------+
|  VIM   |          Result           |
+--------+---------------------------+
| vimnew | Vim updated successfully |
+--------+---------------------------+
```

template_update_vim.json File

Description of the `template_update_vim.json` file used to update the details of a VIM is already added to VNF-LCM. The file is stored at `/ericsson/vnflcm/data` folder.

12.7.8 Manage VNF in VNF-LCM

This utility allows administrators to perform the following operations:

- Discover VNFs in VNF-LCM and Add them to OSS-RC/ENM on page 159
- Discover VNFs in VNF-LCM Without Adding them to OSS-RC/ENM on page 170
- Add VNFs to the OSS-RC/ENM Topology on page 172
- List VNF Details on page 174
- Query VNF Details on page 175
- Get Operational State of a VNF on page 179
- Change Operational State of a VNF on page 180
- Change Operational State of a VNF on page 180
- Delete VNF from VNF-LCM and OSS-RC/ENM Topology on page 182
- Delete VNF from OSS-RC/ENM Topology on page 184

12.7.8.1 Discover VNFs in VNF-LCM and Add them to OSS-RC/ENM

VNFs that are instantiated outside of the VNF-LCM can be discovered in VNF-LCM and OSS-RC/ENM. After the VNFs are discovered, additional LCM operations can be performed on them using VNF-LCM.

The VNFs are discovered using a JSON input file. Multiple VNFs can be discovered in VNF-LCM and added to OSS-RC/ENM either from the same VIM or from different VIMs. If the discovery of one VNF fails during the operation, it does not affect the other VNFs.

Syntax:

The following syntax is used to discover an instantiated VNFs:

```
vnflcm vnf add [-h] [--file FILE]
```

Table 43 Argument Description

Argument	Description
`-h, --help`	Show this help message and exit
`--file FILE`	The JSON file used to discover the VNF in VNF-LCM and OSS-RC/ENM.

Prerequisites

— One or more VNFs that are instantiated outside of the VNF-LCM and have not already been discovered in the VNF-LCM.

— VIMs of the corresponding VNFs must be registered in VNF-LCM using add VIM utility described in section Add a New VIM on page 148.

— ENM/OSS-RC must be configured in VNF-LCM.

— VNF Package manually on-boarded to VNF-LCM.

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server is running.

— Basic understanding of JSON format.

— A backup of the `template_vnf.json` file, taken before executing this procedure.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Fetch vappId.

   Refer to section Fetching vappId from the Cloud on page 185 for more details.
3. Take a backup of original file `template_vnf.json` and name the new file to `vnf.json`.

   ```
   [root@vnflafservices-0 tmp]# cp /ericsson/vnflcm/data/
   template_vnf.json /tmp/vnf.json
   ```
4. Edit the `vnf.json` file to update its properties present under `/tmp`.

   ```
   [root@vnflafservices-0 tmp]# vi vnf.json
   ```

   Refer to section JSON Templates on page 186 to learn more about VNF Template file (`template_vnf.json`).
5. Run the `Discover VNF` command.

   ```
   [root@vnflafservices-0 tmp]# vnflcm vnf add --file=/tmp/vnf.json
   ```

   #### Example

   a. Discover a Single VNF to the VNF-LCM and OSS-RC/ENM.

      The following example shows a single VNF being discovered in VNF-LCM and OSS-RC/ENM :

```
[root@vnflafservices-0 tmp]# vnflcm vnf add --file vnf.json
Adding new vnf: vEPGtest
Starting OSS topology update for vnf vEPGtest
Oss topology updated with the vnf vEPGtest
VNF FDN value updation completed for vEPGtest
+--------------------------------------+--------------------------  →
-----------+------------------+-----------------------------------  →
--+
|                VnfId                 |               Vapp_Id       →
          | Operation_Status |               Message               →
  |
+--------------------------------------+--------------------------  →
-----------+------------------+-----------------------------------  →
--+
| 34469cfe-06f1-11eb-a38f-fa163ef231d7 | be411bd5-9078-42c3-9bb3-64  →
5d29abedd5 |    Successful    | Update to EVNFM and OSS successful  →
  |
+--------------------------------------+--------------------------  →
-----------+------------------+-----------------------------------  →
--+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_152834  →
.json
```

Following is the Input File (vnf.json) used for discovering VNF:

```
{
    "vimconnections": [{
        "vimconnectionId": "01",
        "vimName": "vim1",
        "userDomain": "",
        "tenantName": "vim1_tenant1",
        "subtenantName": ""
    }],
    "vnfs": [{
        "vimconnectionId": "01",
        "vnfInstanceName": "vEPGtest",
        "vappName": "vEPGtest",
        "vappId": "be411bd5-9078-42c3-9bb3-645d29abedd5",
        "vnfVersion": "16A",
        "vnfType": "EPG",
        "addVnfToOss": "true",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "instantiated from vnf",
            "onboardedVnfPkgInfoId": "",
            "vnfdId": "vnfId",
            "flavourId": "m1.medium",
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
            },
            "metadata": {},
            "extensions": {},
            "fdn": "",
            "scalingByMoreThanOneStepSupported": "true"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "19.1.1.1",
            "nodeUsername": "root",
            "nodePassword": "shroot",
            "communityString": "",
            "subNetworks": "Mvnf",
            "netConfPort": "",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
```

```
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "True",
            "associatedSite": "LMC",
            "managedElementId": "vEPGtest",
            "networkElementVersion": "16A",
            "managedElementSrcType": "SSR",
            "managedElementTypes": "Vinfra",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }, {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }]
        }
    }]
}
```

b. Discover multiple VNFs from same VIM

The following example shows a multiple VNFs from a single VIM being discovered in VNF-LCM and OSS-RC/ENM.

```
[root@vnflafservices-0 tmp]# vnflcm vnf add --file add_dual_vnf.jso →
n
Adding new vnf: vEPGtest1
Starting OSS topology update for vnf vEPGtest1
Oss topology updated with the vnf vEPGtest1
VNF FDN value updation completed for vEPGtest1
Adding new vnf: vEPGtest2
Starting OSS topology update for vnf vEPGtest2
Oss topology updated with the vnf vEPGtest2
VNF FDN value updation completed for vEPGtest2
+--------------------------------------+--------------------------- →
-----------+-----------------+----------------------------------- →
--+
|                VnfId                 |              Vapp_Id       →
           | Operation_Status |               Message              →
  |
+--------------------------------------+--------------------------- →
-----------+-----------------+----------------------------------- →
--+
| 34469cfe-06f1-11eb-a38f-fa163ef231d7 | be411bd5-9078-42c3-9bb3-64 →
5d29abedd5 |    Successful    | Update to EVNFM and OSS successful →
  |
| 3f2de760-06f1-11eb-a38f-fa163ef231d7 | 91054eee-69ee-4d07-8916-2a →
9d991289c7 |    Successful    | Update to EVNFM and OSS successful →
  |
+--------------------------------------+--------------------------- →
-----------+-----------------+----------------------------------- →
--+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_152834 →
.json
```

Following is the Input File (`vnf.json`) used for discovering VNF:

```
{
    "vimconnections": [{
        "vimconnectionId": "01",
        "vimName": "cee01",
        "userDomain": "",
        "tenantName": "VNF-LCM",
        "subtenantName": ""
    }],
```

```
    "vnfs": [{
        "vimconnectionId": "01",
        "vnfInstanceName": "vEPGtest1",
        "vappName": "vEPGtest1",
        "vappId": "be411bd5-9078-42c3-9bb3-645d29abedd5",
        "vnfVersion": "16B-R13C",
        "vnfType": "EPG",
        "addVnfToOss": "true",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "vEPGtest 1",
            "vnfdId": "vnflaf",
            "flavourId": "m1.medium",
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
            },
            "metadata": {},
            "extensions": {},
            "fdn": "",
            "scalingByMoreThanOneStepSupported": "true"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "10.11.21.29",
            "nodeUsername": "root",
            "nodePassword": "shroot",
            "communityString": "",
            "subNetworks": "",
            "netConfPort": "830",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "",
            "associatedSite": "LMC",
            "managedElementId": "TestEPG1",
            "netWorkElementVersion": "16B-R13C",
            "managedElementSrcType": "SSR",
            "managedElementTypes": "Vinfra",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }, {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }]
        }
    }, {
        "vimconnectionId": "01",
        "vnfInstanceName": "vEPGtest2",
        "vappName": "vEPGtest2",
        "vappId": "91054eee-69ee-4d07-8916-2a9d991289c7",
        "vnfVersion": "16B-R13C",
        "vnfType": "EPG",
        "addVnfToOss": "false",
        "queryManagedElementId": "TestEPG2",
        "PersistVnf": {
            "vnfInstanceDescription": "VNF EPG 2",
            "vnfdId": "vnflaf1",
            "flavourId": "m1.medium",
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
```

```
            },
            "metadata": {},
            "extensions": {},
            "fdn": "MeContext=TestEPG2",
            "scalingByMoreThanOneStepSupported": "true"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ]
    }]
}
```

c. Discover multiple VNFs from different VIMs.

The following example shows multiple VNFs from the different VIMs being discovered in VNF-LCM and ENM. As seen in the output, the VNFs are discovered one by one. The result is displayed when the operation is complete for all VNFs. The output is saved in the file even if one VNF fails to be discovered.

```
[root@vnflafservices-0 tmp]# vnflcm vnf add --file vnf →
.json
Adding New vnf: AndromedaDND in persist
VNF FDN value updation completed for AndromedaDND
Adding New vnf: zvaxapa_5015_dnd in persist
VNF FDN value updation completed for zvaxapa_5015_dnd
Adding New vnf: zvaxapa_5015_dnd in persist
Error in add_persistvnf operation. Failed to update vn →
fInstance in vnflaf-db. Refer logs at /var/log/vnflcm- →
admin-cli for details
Adding New vnf: taf_test_dnd in persist
VNF FDN value updation completed for taf_test_dnd
+--------------------------------------+-------------- →
-----------------------+----------------------------- →
+-------------------------------------------------------+
| VnfId                                | Vapp_Id       →
                       | Operation_Status              →
| Message                                               |
+--------------------------------------+-------------- →
-----------------------+----------------------------- →
+-------------------------------------------------------+
| a5d38761-132d-11e9-8333-fa163ee542b7 | 0e5e1303-8ea9 →
-4121-a09d-a13e0c21babc | Successful                  →
| Update to EVNFM and OSS successful                    |
| a8498f8a-132d-11e9-8333-fa163ee542b7 | 788badd9-c359 →
-4652-8742-07395e2465d3 | Successful                  →
| Update to EVNFM and OSS successful                    |
| 788badd9-c359-4652-8742-07395e2465d3 | Update to DB  →
and OSS failed          | -                             →
| VNF with provided vAppId already exists in EVNFM      |
| aad829ca-132d-11e9-8333-fa163ee542b7 | 70f2bfcb-7ada →
```

```
-4129-b985-1552e8f6c5a0 | Successful                    →
| Update to EVNFM and OSS successful                |
+-------------------------------------+-------------- →
-----------------------+------------------------------ →
+-------------------------------------------------+
Output saved in file /var/log/vnf_discovery/add_vnf_20 →
190114_111301.json

Below is the Input File (vnf.json) used for discoverin →
g VNF:
{
    "vimconnections": [{
        "vimconnectionId": "01",
        "vimName": "cee01",
        "userDomain": "",
        "tenantName": "VNF-LCM",
        "subtenantName": ""
    }, {
        "vimconnectionId": "02",
        "vimName": "cee02",
        "userDomain": "",
        "tenantName": "NFV",
        "subtenantName": ""
    }],
    "vnfs": [{
        "vimconnectionId": "01",
        "vnfInstanceName": "AndromedaDND",
        "vappName": "AndromedaDND",
        "vappId": "0e5e1303-8ea9-4121-a09d-a13e0c21bab →
c",
        "vnfVersion": "16B-R13C",
        "vnfType": "EPG",
        "addVnfToOss": "false",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "VNF 1",
            "vnfdId": "vnflaf",
            "flavourId": "m1.medium",
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
            },
            "metadata": {},
            "extensions": {},
            "fdn": "MeContext=TestEPG1",
            "scalingByMoreThanOneStepSupported": "true →
"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
```

```
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "",
            "nodeUsername": "",
            "nodePassword": "",
            "communityString": "",
            "subNetworks": "",
            "netConfPort": "",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "",
            "associatedSite": "",
            "managedElementId": "",
            "networkElementVersion": "",
            "managedElementSrcType": "",
            "managedElementTypes": "",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }, {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }]
        }
    }, {
        "vimconnectionId": "01",
        "vnfInstanceName": "zvaxapa_5015_dnd",
        "vappName": "zvaxapa_5015_dnd",
        "vappId": "788badd9-c359-4652-8742-07395e2465d →
3",
        "vnfVersion": "16B-R13C",
        "vnfType": "EPG",
        "addVnfToOss": "false",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "VNF 2",
            "vnfdId": "vnflaf1",
            "flavourId": "m1.medium",
```

```
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
            },
            "metadata": {},
            "extensions": {},
            "fdn": "MeContext=TestEPG2",
            "scalingByMoreThanOneStepSupported": "true →
"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "",
            "nodeUsername": "",
            "nodePassword": "",
            "communityString": "",
            "subNetworks": "",
            "netConfPort": "",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "",
            "associatedSite": "",
            "managedElementId": "",
            "networkElementVersion": "",
            "managedElementSrcType": "",
            "managedElementTypes": "",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }, {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }]
        }
```

```
    }, {
        "vimconnectionId": "01",
        "vnfInstanceName": "zvaxapa_5015_dnd",
        "vappName": "zvaxapa_5015_dnd",
        "vappId": "788badd9-c359-4652-8742-07395e2465d →
3",
        "vnfVersion": "16B-R13C",
        "vnfType": "EPG",
        "addVnfToOss": "false",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "king of the jun →
gle",
            "vnfdId": "vnflaf1",
            "flavourId": "m1.medium",
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
            },
            "metadata": {},
            "extensions": {},
            "fdn": "MeContext=TestEPG2",
            "scalingByMoreThanOneStepSupported": "true →
"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "",
            "nodeUsername": "",
            "nodePassword": "",
            "communityString": "",
            "subNetworks": "",
            "netConfPort": "",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "",
            "associatedSite": "",
            "managedElementId": "",
```

```
            "networkElementVersion": "",
            "managedElementSrcType": "",
            "managedElementTypes": "",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }, {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }]
        }
    }, {
        "vimconnectionId": "02",
        "vnfInstanceName": "taf_test_dnd",
        "vappName": "taf_test_dnd",
        "vappId": "70f2bfcb-7ada-4129-b985-1552e8f6c5a →
0",
        "vnfVersion": "16B-R13C",
        "vnfType": "EPG",
        "addVnfToOss": "false",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "VNF 3",
            "vnfdId": "vnflaf3",
            "flavourId": "m1.medium",
            "vnfConfigurableProperties": {
                "isAutoScaleEnabled": "true",
                "isAutoHealingEnabled": "true"
            },
            "metadata": {},
            "extensions": {},
            "fdn": "MeContext=TestEPG3",
            "scalingByMoreThanOneStepSupported": "true →
"
        },
        "scaleAspectInfo": [{
                "id": "GPB",
                "currentScaleLevel": 1
            },
            {
                "id": "vLC",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "",
            "nodeUsername": "",
            "nodePassword": "",
```

```
            "communityString": "",
            "subNetworks": "",
            "netConfPort": "",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "",
            "associatedSite": "",
            "managedElementId": "",
            "networkElementVersion": "",
            "managedElementSrcType": "",
            "managedElementTypes": "",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }, {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": ""
            }]
        }
    }]
}
```

As seen in the output, the VNFs are added one by one. The overall final result displays once the operation is completed for all VNFs. The display of the final result occurs regardless of the failure of operation for any VNF.

12.7.8.2 Discover VNFs in VNF-LCM Without Adding them to OSS-RC/ENM

VNFs that are instantiated outside of the VNF-LCM can be discovered in VNF-LCM. Details of the VNFs to be discovered in VNF-LCM needs to be provided in a JSON input file. Multiple VNFs can be discovered in VNF-LCM either from the same VIM or from different VIMs. Failure to discover one VNF does not affect the other VNFs.

Syntax:

The following syntax is used to discover existing VNFs in VNF-LCM.

```
vnflcm vnf add_persistvnf [-h] [--file FILE]
```

Table 44 Argument Description

Argument	Description
`-h, --help`	Show this help message and exit
`--file FILE`	The JSON file used to discover the VNF in VNF-LCM and OSS-RC/ENM.

Prerequisites

- One or more VNFs that are instantiated outside of the VNF-LCM and have not discovered in VNF-LCM.
- VIMs of the corresponding VNFs must be registered in VNF-LCM using add VIM utility described in section: 9.7.1 Add a New VIM to VNF-LCM Framework.
- VNF Package manually on-boarded to VNF-LCM.
- Root and cloud-user access to the vnflaf-services VM.
- Jboss server is running.
- Basic understanding of JSON format.
- A backup of the `template_vnf.json` file taken before running this procedure.

Steps

1. Log on to the service pod.

   ```
   kubectl exec -it <servicepod> -n <namespace> /bin/bash
   ```

   Example

   ```
   kubectl exec -it eric-vnflcm-service-0 -n exp /bin/bash
   ```

2. Fetch the vappId.

   Refer to section Fetching vappId from the Cloud on page 185 for more details.

3. Copy the `template_vnf.json` file and name the new file `vnf.json`.

   ```
   eric-vnflcm-service-0:/ # cp /ericsson/vnflcm/data/template_vnf.json /tmp/vn →
   f.json
   ```

4. Update the properties in the `vnf.json` file, which is inside `/tmp`.

   ```
   eric-vnflcm-service-0:/ # vi vnf.json
   ```

Refer to section JSON Templates on page 186 to learn more about the VNF Template file (template_vnf.json).

5. Run the `Discover VNF` command.

```
eric-vnflcm-service-0:/ # vnflcm vnf add_persistvnf --file=/tmp/vnf.json
```

Example

```
[root@vnflafservices-0 tmp]# vnflcm vnf add_persistvnf --file
 vnf.json
Discover a New vnf: vnf1 in persist
Persist VNF successful for vnf1
+--------------------------------------+--------------------
-----------------+------------------+------------------------
--------------+
|                VnfId                 |              Vapp_I
d               | Operation_Status |              Message
             |
+--------------------------------------+--------------------
-----------------+------------------+------------------------
--------------+
| 34469cfe-06f1-11eb-a38f-fa163ef231d7 | be411bd5-9078-42c3-9
bb3-645d29abedd5 |    Successful    | Update to EVNFM success
ful           |
+--------------------------------------+--------------------
-----------------+------------------+------------------------
--------------+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_
152834.json
```

12.7.8.3 Add VNFs to the OSS-RC/ENM Topology

This utility allows administrator to add VNF(s) to OSS-RC/ENM Topology. Details of the VNFs to be added to OSS-RC/ENM needs to be provided in a JSON input file.

Note: Applicable for ENM: Few VNFs need to follow enrollment procedure (that is, configuring LDAP Credentials) for the VNF to be able to use the ENM LDAP. See Node Certificate Administrative Tasks in ENM Network Security Configuration System Administrator Guide [5].

Syntax:

```
vnflcm vnf add_updateoss [-h] [--file FILE]
```

Arguments:

`-h, --help` show this help message and exit

`--file FILE JSON` file to add vnf details to OSS-RC/ENM topology

Prerequisites

— An existing VNF that is already present in VNF-LCM.

— ENM/OSS-RC must be configured in VNF-LCM.

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server is running

— Basic understanding of JSON format.

— A backup of the `template_vnf.json` file, taken before executing this procedure.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Take a backup of original file `template_vnf.json` and name the new file to `vnf.json`.

   ```
   [root@vnflafservices-0 tmp]# cp /ericsson/vnflcm/data/
   template_vnf.json /tmp/vnf.json
   ```

3. Edit the `vnf.json` file to update its properties present under `/tmp`.

   ```
   [root@vnflafservices-0 tmp]# vi vnf.json
   ```

   Refer to section JSON Templates on page 186 to learn more about the file.

4. Run 'Add VNF' CLI command:

   ```
   [root@vnflafservices-0 tmp]# vnflcm vnf add_updateoss --file vnf.json
   ```

   #### Example

   ```
   [root@vnflafservices-0 tmp]# vnflcm vnf add_updateoss --file vnf.json
   Adding New vnf: vnf1
   Starting OSS topology update for vnf vnf1
   Topology updated successfully for vnf1
   Oss topology updated with the vnf vnf1
   +--------------------------------------+----------------------------------- →
   --+-----------------+--------------------------------------+
   |               VnfId                  |              Vapp_Id               →
     | Operation_Status |              Message                 |
   +--------------------------------------+----------------------------------- →
   --+-----------------+--------------------------------------+
   | 34469cfe-06f1-11eb-a38f-fa163ef231d7 | be411bd5-9078-42c3-9bb3-645d29abedd →
   5 |    Successful    | Update to OSS successful             |
   +--------------------------------------+----------------------------------- →
   ```

```
--+------------------+-------------------------------------+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_152834.json
```

12.7.8.4 List VNF Details

This utility allows administrator to list all the VNFs present in VNF-LCM and also to list the VNFs matching with a given vapp name.

Syntax:

```
vnflcm vnf list [-h] [--vappname VAPPNAME]
```

Arguments:

`-h, --help` show this help message and exit

--vappname VAPPNAME vApp/stack Name in the Cloud

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server is running

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Run list command:

```
[root@vnflaf-services ~]# vnflcm vnf list [--vappname <VAPP_NAME>]
```

Example

List all VNFs:

```
[root@vnflafservices-0 tmp]# vnflcm vnf list
+-------+--------------------------------------+--------------------------- →
----------+----------+---------+------------+--------------------+---------- →
-------------+
| index |                vnfId                 |                vappId       →
          | vappName | vnfType | vnfVersion | instantiationState | vnfOperat →
ionStateType |
+-------+--------------------------------------+--------------------------- →
----------+----------+---------+------------+--------------------+---------- →
-------------+
|   1   | 5c7028c6-12ca-11eb-a2e7-9a7e369471f8 |                 None        →
          |   None   |   None  |   18.08    |  NOT_INSTANTIATED  |           →
None         |
|   2   | e6b5cc4d-12d5-11eb-b414-9a7e369471f8 | b3d99bab-c16a-485c-ba4f-7f9 →
711c57414 | TestVNF1 |   None  |   18.08    |    INSTANTIATED    |     ADMIN →
_UNLOCKED    |
|   3   | afe4f0dc-1375-11eb-b414-9a7e369471f8 |                 None        →
```

```
          |   None   |   None  |   18.08    |  NOT_INSTANTIATED  |
None         |
|   4   | 14ef1cec-1378-11eb-b414-9a7e369471f8 |                None
          |   None   |   None  |   18.08    |  NOT_INSTANTIATED  |
None         |
+-------+--------------------------------------+--------------------------
----------+----------+---------+------------+--------------------+----------
-------------+
VNF details listed successfully
```

List VNF by vappName

```
[root@vnflafservices-0 tmp]# vnflcm vnf list --vappname TestVNF1
+-------+--------------------------------------+--------------------------
----------+----------+---------+------------+--------------------+----------
-------------+
| index |                vnfId                 |               vappId
          | vappName | vnfType | vnfVersion | instantiationState | vnfOperat
ionStateType |
+-------+--------------------------------------+--------------------------
----------+----------+---------+------------+--------------------+----------
-------------+
|   1   | e6b5cc4d-12d5-11eb-b414-9a7e369471f8 | b3d99bab-c16a-485c-ba4f-7f9
711c57414 | TestVNF1 |   None  |   18.08    |    INSTANTIATED    |     ADMIN
_UNLOCKED    |
+-------+--------------------------------------+--------------------------
----------+----------+---------+------------+--------------------+----------
-------------+
```

12.7.8.5 Query VNF Details

This utility allows the administrators to lists the details of an instantiated VNF using vnfid or vnfname.

Syntax:

```
vnflcm vnf query_vnf_details [-h] [--vnfid VNFID] [--vnfname VNF
NAME]
```

Arguments:

`-h, --help` show this help message and exit

`--vnfid` VNFID VNF Instance ID

`--vnfname`VNFNAME VNF Instance Name

Note: VNFID and VNFNAME are optional parameters, if not provided, the result has details of all the VNFs instantiated.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server is running

— An existing VNF that is already instantiated and added to VNF-LCM.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Execute the Query VNF details command to get the details of the VNFs:

```
[root@vnflaf-services ~]# vnflcm vnf query_vnf_details --vnfid <vnfId>
```

Example

Query VNF Details using vnfid:

```
[root@vnflafservices-0 tmp]# vnflcm vnf query_vnf_details --vnfid e6b5cc4d-1 →
2d5-11eb-b414-9a7e369471f8
Query VNF details call was successful
[
    {
        "vnfInstanceName": "TestVNF1",
        "vnfProductName": "VNFLAF",
        "vnfInstanceDescription": "VNF creation using example package",
        "instantiationState": "INSTANTIATED",
        "vimConnectionInfo": [
            {
                "interfaceInfo": {
                    "identityEndPoint": "https://cloud12a.athtem.eei.ericsso →
n.se:13000/v3"
                },
                "vimType": "OPENSTACK",
                "accessInfo": {
                    "projectId": "795f6406875b41bcb277d13cc8785c6f",
                    "domainName": "Default",
                    "projectName": "ORCH_VNFLCM_Flash_C12A15_Dynamic"
                },
                "id": "ed5da14c-b90d-43d0-8745-710e24f45f92",
                "vimId": "ieatdoxb032_cloud12a"
            }
        ],
        "vnfProvider": "Ericsson",
        "instantiatedVnfInfo": {
            "extManagedVirtualLinkInfo": [],
            "vnfVirtualLinkResourceInfo": [],
            "extVirtualLinkInfo": [
                {
                    "resourceHandle": {
                        "resourceId": "226cde9c-619d-4d6c-a1e8-e1e565723658" →
,
                        "vimLevelResourceType": "OS::Neutron::Net",
                        "vimConnectionId": "ed5da14c-b90d-43d0-8745-710e24f4 →
5f92"
                    },
                    "id": "226cde9c-619d-4d6c-a1e8-e1e565723658",
                    "extLinkPorts": []
                }
            ],
            "extCpInfo": [],
            "flavourId": "cee",
            "vnfState": "STARTED",
            "vnfcResourceInfo": [
                {
                    "storageResourceIds": [],
                    "computeResource": {
                        "resourceId": "6eed9341-fa27-46e4-b299-6cbe52544307" →
,
                        "vimLevelResourceType": "OS::Nova::Server",
                        "vimConnectionId": "ed5da14c-b90d-43d0-8745-710e24f4 →
5f92"
                    },
                    "vduId": "vnflafecm-services.0.vnflafecm-services",
```

```
                    "resourceName": "vnflafecm-services.0.vnflafecm-services
",
                    "vnfcCpInfo": [
                        {
                            "cpProtocolInfo": [
                                {
                                    "layerProtocol": "IP_OVER_ETHERNET",
                                    "ipOverEthernet": {
                                        "macAddress": "fa:16:3e:cf:76:ad",
                                        "ipAddresses": [
                                            {
                                                "subnetId": "d1bf0278-5770-4
dfb-8ca4-cbd94b8650af",
                                                "type": "IPV4",
                                                "addresses": [
                                                    "10.232.8.18"
                                                ]
                                            }
                                        ]
                                    }
                                }
                            ],
                            "vnfExtCpId": "vnflafecm-services.0.vnflaf_servi
ces_external_port",
                            "id": "vnflafecm-services.0.vnflaf_services_exte
rnal_port",
                            "cpdId": "vnflafecm-services.0.vnflaf_services_e
xternal_port"
                        }
                    ],
                    "id": "VNFLAF-Services-0"
                }
            ],
            "virtualStorageResourceInfo": [],
            "scaleStatus": [
                {
                    "scaleLevel": 0,
                    "aspectId": "VNFLAF-Services"
                }
            ]
        },
        "vnfdId": "vnflafecm",
        "extensions": {},
        "vnfdVersion": "1.0",
        "vnfConfigurableProperties": {
            "isAutoHealingEnabled": "true",
            "isAutoScaleEnabled": "true"
        },
        "_links": {
            "self": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8"
            },
            "scale": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8/scale"
            },
            "terminate": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8/terminate"
            },
            "heal": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8/heal"
            }
        },
        "vnfPkgId": "e6a18b78-b4c5-4d33-aba4-9e1ddde1fb46",
        "id": "e6b5cc4d-12d5-11eb-b414-9a7e369471f8",
        "vnfSoftwareVersion": "18.08",
        "metadata": {}
    }
]
```


Query VNF Details using vnfname:

```
[root@vnflafservices-0 tmp]# vnflcm vnf query_vnf_details --vnfname "TestVNF
1"
```

```
Query VNF details call was successful
[
    {
        "vnfInstanceName": "TestVNF1",
        "vnfProductName": "VNFLAF",
        "vnfInstanceDescription": "VNF creation using example package",
        "instantiationState": "INSTANTIATED",
        "vimConnectionInfo": [
            {
                "interfaceInfo": {
                    "identityEndPoint": "https://cloud12a.athtem.eei.ericsso →
n.se:13000/v3"
                },
                "vimType": "OPENSTACK",
                "accessInfo": {
                    "projectId": "795f6406875b41bcb277d13cc8785c6f",
                    "domainName": "Default",
                    "projectName": "ORCH_VNFLCM_Flash_C12A15_Dynamic"
                },
                "id": "ed5da14c-b90d-43d0-8745-710e24f45f92",
                "vimId": "ieatdoxb032_cloud12a"
            }
        ],
        "vnfProvider": "Ericsson",
        "instantiatedVnfInfo": {
            "extManagedVirtualLinkInfo": [],
            "vnfVirtualLinkResourceInfo": [],
            "extVirtualLinkInfo": [
                {
                    "resourceHandle": {
                        "resourceId": "226cde9c-619d-4d6c-a1e8-e1e565723658" →
,
                        "vimLevelResourceType": "OS::Neutron::Net",
                        "vimConnectionId": "ed5da14c-b90d-43d0-8745-710e24f4 →
5f92"
                    },
                    "id": "226cde9c-619d-4d6c-a1e8-e1e565723658",
                    "extLinkPorts": []
                }
            ],
            "extCpInfo": [],
            "flavourId": "cee",
            "vnfState": "STARTED",
            "vnfcResourceInfo": [
                {
                    "storageResourceIds": [],
                    "computeResource": {
                        "resourceId": "6eed9341-fa27-46e4-b299-6cbe52544307" →
,
                        "vimLevelResourceType": "OS::Nova::Server",
                        "vimConnectionId": "ed5da14c-b90d-43d0-8745-710e24f4 →
5f92"
                    },
                    "vduId": "vnflafecm-services.0.vnflafecm-services",
                    "resourceName": "vnflafecm-services.0.vnflafecm-services →
",
                    "vnfcCpInfo": [
                        {
                            "cpProtocolInfo": [
                                {
                                    "layerProtocol": "IP_OVER_ETHERNET",
                                    "ipOverEthernet": {
                                        "macAddress": "fa:16:3e:cf:76:ad",
                                        "ipAddresses": [
                                            {
                                                "subnetId": "d1bf0278-5770-4 →
dfb-8ca4-cbd94b8650af",
                                                "type": "IPV4",
                                                "addresses": [
                                                    "10.232.8.18"
                                                ]
                                            }
                                        ]
                                    }
                                }
                            ],
                            "vnfExtCpId": "vnflafecm-services.0.vnflaf_servi →
ces_external_port",
                            "id": "vnflafecm-services.0.vnflaf_services_exte →
rnal_port",
```

```
                            "cpdId": "vnflafecm-services.0.vnflaf_services_e
xternal_port"
                        }
                    ],
                    "id": "VNFLAF-Services-0"
                }
            ],
            "virtualStorageResourceInfo": [],
            "scaleStatus": [
                {
                    "scaleLevel": 0,
                    "aspectId": "VNFLAF-Services"
                }
            ]
        },
        "vnfdId": "vnflafecm",
        "extensions": {},
        "vnfdVersion": "1.0",
        "vnfConfigurableProperties": {
            "isAutoHealingEnabled": "true",
            "isAutoScaleEnabled": "true"
        },
        "_links": {
            "self": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8"
            },
            "scale": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8/scale"
            },
            "terminate": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8/terminate"
            },
            "heal": {
                "href": "https://evnfm.ccd-c7a020.athtem.eei.ericsson.se/vnf
lcm/v1/vnf_instances/e6b5cc4d-12d5-11eb-b414-9a7e369471f8/heal"
            }
        },
        "vnfPkgId": "e6a18b78-b4c5-4d33-aba4-9e1ddde1fb46",
        "id": "e6b5cc4d-12d5-11eb-b414-9a7e369471f8",
        "vnfSoftwareVersion": "18.08",
        "metadata": {}
    }
]
```


12.7.8.6 Get Operational State of a VNF

This utility allows administrator to fetch the Operational State of a VNF using vnfid or vnfname.

Syntax:

```
vnflcm vnf get_operational_state [-h] [--vnfid VNFID] [--vnfname
 VNFNAME]
```

Arguments:

`-h, --help` show this help message and exit

`--vnfid` VNFID VNF Instance ID

`--vnfname`VNFNAME VNF Instance Name

Prerequisites

— Root and cloud-user access to the vnflaf VM.

— Jboss/Server must be running.

— An existing VNF that is already instantiated and added to VNF-LCM.

Steps

1. Log on to vnflcm VM as cloud-user, then switch to root user.
2. Fetch the operational state of VNF through VNFID or VNFNAME.

```
[root@vnflaf-services ~]# vnflcm vnf get_operational_state --vnfid <vnfId>
```

Example

Get VNF operational state using vnfid:

```
[root@vnflafservices-0 tmp]# vnflcm vnf get_operational_state --vnfid "8194a →
0d0-2073-11e9-b0a5-fa163e532e93"

VNF is currently in ADMIN_LOCKED operational state
```

Get VNF operational state using vnfname:

Example

```
[root@vnflafservices-0 tmp]# vnflcm vnf get_operational_state --vnfname "vnf →
lcm123"

VNF is currently in ADMIN_LOCKED operational state
```

12.7.8.7 Change Operational State of a VNF

This utility allows administrator to change the Operational State of a VNF using vnfid or vnfname.

Syntax:

```
vnflcm vnf change_operational_state [-h] [--vnfid VNFID] [--vnfname VNFNAME] [-- →
changestateto CHANGESTATETO]
```

Arguments:

`-h, --help` show this help message and exit

`--vnfid` VNFID VNF Instance ID

`--vnfname`VNFNAME VNF Instance Name

`--changestateto` CHANGESTATETO Operational State of the VNF. Accepted Values : `{"ADMIN_LOCKED", "ADMIN_UNLOCKED"}`

Prerequisites

— Root and cloud-user access to the vnflaf VM.

— Jboss/Server must be running.

— An existing VNF that is already instantiated and added to VNF-LCM.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Change the operational state of VNF through VNFID or VNFNAME, and the value to be updated through `CHANGESTATETO`:

```
[root@vnflafservices-0 tmp]# vnflcm vnf change_vnf_operational_state --vnfid
 <vnfId> --changestateto "<value to be updated>"
```

Example

Following is the example for changing VNF operation state using vnfid:

```
[root@vnflafservices-0 tmp]# vnflcm vnf change_vnf_operational_state --vnfid
 "8194a0d0-2073-11e9-b0a5-fa163e532e93" --changestateto "ADMIN_UNLOCKED"

VNF Operational State successfully changed to ADMIN_UNLOCKED
```

Example

Following is the example for changing VNF operation state using vnfname

```
[root@vnflafservices-0 tmp]# vnflcm vnf change_operational_state --vnfname "
vnflcm123" --changestateto "ADMIN_UNLOCKED"

VNF Operational State successfully changed to ADMIN_UNLOCKED
```

12.7.8.8 Update VNF Scaling Info

This admin CLI utility allows updating scaling info in the VNFs already added to VNF-LCM which does not support scale to enable scale operations.

Syntax:

```
vnflcm vnf update_scale_info [-h] [--file FILE]
```

Arguments:

`-h, --help` show this help message and exit

`--file` FILE JSON file to update vnf scale info

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server must be running.

— An existing VNF that is already instantiated (without using VNF-LCM) and added to VNF-LCM.

— Basic understanding of JSON format.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Edit the `template_scale_update.json` to update its properties present under `/ericsson/vnflcm/data/`.

```
[root@vnflaf-services data] vi template_scale_update.json
```

Refer to section JSON Templates on page 186 to learn more about `template_scale_update.json`

3. Execute the command:

```
[root@vnflaf-services ]# vnflcm vnf update_scale_info --file <sclaleInfo_upd →
ate.json>
```

Example

```
[root@vnflafservices-0 tmp]# vnflcm vnf update_scale_info --file template_sc →
ale_update.json
+--------+-------------------------------------------------+
| VNF       |                            Result                          →
      |
+--------+-------------------------------------------------+
| mvnf37 | VNF is successfully updated with scale info    |
+--------+-------------------------------------------------+
```

12.7.8.9 Delete VNF from VNF-LCM and OSS-RC/ENM Topology

This utility is provided in cases where the workflow fails to clean up the failed VNF instances. This can be used to delete the failed/cancelled VNF instances from VNF-LCM database by providing the VNF instance name irrespective of VNF's instantiated or non-instantiated state. It is assumed that the resources occupied by that VNF has already been released manually and EO database is already in sync with that. If any operation ongoing on the requested VNF to delete, the user will not be allowed to delete that instance. Rollback is not supported in context of this utility. Resource rollback should already be handled by workflow.

Syntax:

```
vnflcm vnf delete [-h] [--name NAME]
```

Arguments:

`-h, --help` show this help message and exit

`--name` NAME Name of VNF

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server is running.

— VNF to be deleted exists in VNF-LCM DB.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Delete the VNF through VNF Name, which is in failed/cancelled state:

   ```
   [root@vnflaf-services ~]# vnflcm vnf delete --name <vnfName>
   ```

   Prior to deletion, user is prompted with a confirmation message:

   ```
   VNF might be in INSTANTIATED or NOT_INSTANTIATED state, are you sure to dele →
   te this? (y/n)
   ```

   ### Example:

   ```
   [root@vnflafservices-0 tmp]# vnflcm vnf delete --name vnf123
   VNF might be in INSTANTIATED or NOT_INSTANTIATED state, are you sure to dele →
   te this? (y/n) y
   VNF is deleted successfully
   ```

   If the VNF is in processing state, user cannot delete the VNF, and prompted with a message as shown in the following:

   ```
   [root@vnflafservices-0 tmp]# vnflcm vnf delete --name vnf12
   VNF might be in INSTANTIATED or NOT_INSTANTIATED state, are you sure to dele →
   te this? (y/n) y
   Unable to delete VNF with name vnf12. The VNF Operation is in processing sta →
   te.
   ```

12.7.8.9.1 Possible Error Scenarios

Error Code	Error Message
404	The VNF identifier is not found in the application.
409	The VNF Operation is in processing state.
403	The VNF has associated resources.
500	Unable to process the request by application for vnfname <vnfName>.This is an internal system error. Refer to logs at `/var/log/vnflcm-admin-cli/logfile.log` for details. .

12.7.8.10 Delete VNF from OSS-RC/ENM Topology

This utility allows administrator to delete VNF from OSS/ENM topology by providing the name of the network Managed Object (MO).

Syntax:

```
vnflcm vnf delete_from_oss [-h] [--name NAME]
```

Arguments:

`-h, --help` show this help message and exit

`--name` NAME Name of Network MO, example: NetworkElement=vnf-1

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Jboss/Server is running

— VNF to be deleted exists in OSS/ENM

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. FDelete the VNF using name of network MO:

```
[root@vnflafservices-0 tmp]# vnflcm vnf delete_from_oss --nam →
e <MOName>
```

Example

```
[root@vnflafservices-0 tmp]# vnflcm vnf delete_from_oss --name MeContext=Tes →
t
+-------------------+-------------------+-------------------------------+
| networkMO          | Operation_Status  | Message                       →
      |
+-------------------+-------------------+-------------------------------+
| MeContext=Test   | Success                | Delete from ENM successful |
+------------------+--------------------+-------------------------------+
```

12.7.8.11 Fetching vappId from the Cloud

This section describes steps to fetch 'vappId' referred in table 'VNF Properties' from a specific cloud.

12.7.8.11.1 Fetching vappId for CEE

This section describes steps to fetch 'vappId' referred in table 'VNF Properties' from CEE environment

Prerequisites

— An existing VNF, instantiated without using VNF-LCM.

Steps

1. Log in to the CEE cloud dashboard url.
2. Select **Project** as the **View**.
3. Select the stack in use, then look for the stack **Overview**.

   Under the stack **Overview - Stack Information**, the value of `id` is the `vappId` value.

Results

`vappId` (stackid) value available for use.

12.7.8.11.2 Fetching vappId for OpenStack

This section describes steps to fetch 'vappId' referred in table 'VNF Properties' from OpenStack environment.

Prerequisites

— An existing VNF, instantiated without using VNF-LCM.

Steps

1. Log in to OpenStack Cloud dashboard URL.
2. Click the **Orchestration** tab and select **Stacks**.

   This tab lists the stacks present in the Cloud.
3. Select the **Stack** in use.

   From the **Overview** tab of the stack, the value of `ID` is the `vappid` value.

Results

`vappId` value is available for use.

12.7.8.11.3 Fetching vappId for VCD

This section describes steps to fetch "vappId' referred in table 'VNF Properties' from VCD environment

Prerequisites

— An existing vApp , instantiated without using VNF-LCM.

Steps

1. Log in to the VCD cloud dashboard url.
2. Select **My Cloud**, then select **vApps**.
3. Select the specific vApp.
4. In the url, vappid is found as `vapp=<vappid>`.

Results

`vappId` value available for use.

12.7.8.12 JSON Templates

12.7.8.12.1 VNF Template for Discovering VNF to VNF-LCM and OSS-RC/ENM Topology

This section describes the VNF template file (`template_vnf.json`) file. This file is used to discover VNF(s) to VNF-LCM and OSS-RC/ENM topology. The template file is stored at `/ericsson/vnflcm/data` folder.

```
{
    "vimconnections": [{
        "vimconnectionId": "",
```

```
        "vimName": "",
        "userDomain": "",
        "tenantName": "",
        "subtenantName": ""
    }],
    "vnfs": [{
        "vimconnectionId": "",
        "nfvoTenant": "",
        "vnfInstanceName": "",
        "vappName": "",
        "vappId": "",
        "vnfVersion": "",
        "vnfType": "",
        "addVnfToOss": "",
        "queryManagedElementId": "",
        "PersistVnf": {
            "vnfInstanceDescription": "",
            "onboardedVnfPkgInfoId": "",
            "vnfdId": "",
            "flavourId": "",
            "vnfConfigurableProperties": {},
            "metadata": {},
            "extensions": {},
            "fdn": "",
            "scalingByMoreThanOneStepSupported": "true"
        },
        "scaleAspectInfo": [{
                "id": "processing",
                "currentScaleLevel": 1
            },
            {
                "id": "database",
                "currentScaleLevel": 1
            }
        ],
        "updateOssTopology": {
            "nodeIpAddress": "",
            "nodeUsername": "",
            "nodePassword": "",
            "communityString": "",
            "subNetworks": "",
            "netConfPort": "",
            "snmpPort": "",
            "snmpVersion": "",
            "snmpSecurityLevel": "",
            "snmpSecurityName": "",
            "snmpAuthProtocol": "",
            "snmpPrivProtocol": "",
            "snmpAuthPassword": "",
            "snmpPrivacyPassword": "",
            "pmFunction": "",
            "cmNodeHeartbeatSupervision": "",
```

```
            "fmAlarmSupervision": "",
            "disableLdapUser": "",
            "associatedSite": "",
            "managedElementId": "",
            "managedElementSrcType": "",
            "managedElementTypes": "",
            "transportProtocol": "",
            "networkElementVersion": "",
            "vnfmId": "",
            "privateNetworkGroup": "",
            "timeZone": "",
            "axe": [{
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": "",
                "isManagerIpAddress": "",
                "fileTransferProtocol": "",
                "ftpTlsServerPort": ""
            },
            {
                "axeIoApplications": "",
                "axeNodeClusterIP": "",
                "axeNodeInterfaceAIp": "",
                "axeNodeInterfaceBIp": "",
                "fileTransferProtocol": "",
                "ftpTlsServerPort": ""
            }]
        }
    }]
}
```

Note:

- If VNF-LCM is deployed as value pack to Ericsson Orchestrator-Cloud Manager, then for all the templates, the following parameters are not applicable and these parameters must not be provided in the json file:
  - queryManagedElementId
  - scaleAspectInfo
  - updateOssTopology
- Value of `addVnfToOss` parameter is always false.

File Structure Explained

Table 45 VIM Connection Properties

Attribute	Required	Description
vimconnectionId	Y	User provided identifier for that vim details block. This parameter is used only in 'VNF Properties table' to map the VNF with its associated VIM details.
vimName	Y	Name of the VIM where the existing VNF is already deployed. The value must be same as provided, while saving the vimConnectionInfo using multivim CLI.
userDomain	N	User Domain configured at cloud platform. Applicable for OpenStack/CEE with v3 Keystone/Identity service.
tenantName	Y	The tenant Name in the Cloud where VNF is instantiated.
subtenantName	N	The sub-tenant Name in the Cloud where VNF is instantiated.

Table 46 VNF Properties

Attribute	Required	Description
vimconnectionId	Y	A valid vimconnectionId specified in vimconnections.
vnfInstanceName	Y	Human-readable name of the VNF instance created.
vappName	Y	The name of the vapp or stack on the VIM.
vappId	Y	The vappid (in VCD) or stack id (in CEE and Openstack) of the VNF on the VIM.
vnfVersion	N	Version of the VNF and an optional attribute. If not provided, it is auto-populated with the VNF version from VNF package.
vnfType	Y	The type of the VNF. For example, vMME, vEPG. For vSTP nodes, provide "STP" for OSSRC and "vIP-STP" for ENM. For vMSC and vMSC high capacity nodes, provide "MSCServer" for OSSRC, "vMSC" and "vMSC-HC" for ENM. For vHLR nodes, provide "HLRServer" for OSSRC and vHLR-FE, HLR-FE, HLR-FE-IS, HLR-FE-BSP for ENM.

Attribute	Required	Description
addVnfToOss	N	If true, VNF-LCM will try to add VNF to ENM/OSSRC topology. Default value is false. In case it is false, the VNF is considered to be already added to ENM/OSSRC and hence, correct fdn value should be provided in fdn input parameter. Else, user can provide queryManagedElementId, which is used to fetch FDN from the ENM/OSSRC. If both fdn and queryManagedElementId are provided, VNF-LCM do not try to fetch the fdn from ENM/OSSRC. If false, and neither of fdn and queryManagedElementId are provided , then VNF is not added to VNF-LCM. FDN is needed for supporting autostart use cases (auto heal, auto scale) and also without FDN, VNF-LCM cannot delete the VNF from ENM/OSSRC topology when terminate is executed. **Note:** If VNF-LCM is deployed as value pack to Ericsson Orchestrator-Cloud Manager, the value of parameter `addVnfToOss` always remains false.
queryManagedElementId	N	Managed element Id of a VNF that is already present in ENM/OSSRC. This parameter will be used to fetch the fdn from the ENM/OSSRC. **Note:** Do not provide this parameter, If VNF-LCM is deployed as value pack to Ericsson Orchestrator-Cloud Manager.
scaleAspectInfo	N	Describes the scaling information. Array of `scaleAspectInfo` may contain 1 or more elements. For details, see table 'VNF scaleAspectInfo Properties'. **Note:** For vmWare vCloud Director deployed VNFs (except vMSC Compact and High Capacity nodes), scale related data must be appended in "metadata" field. Example: "metadata": { "aspectIdList":"default"}.

Table 47 VNF PersistVnf properties

Attribute	Required	Description
vnfInstanceDescription	-	Human-readable description of the VNF instance created.
vnfdId	Y	VNF descriptor Id. Identifier to identify the VNF package. For small stack workflows, this is a `packageId` which can be retrieved by `ValidatePackage` reusable workflow.
flavourId	Y	The VNF flavour id.
vnfConfigurableProperties	N	The configurable properties of the VNF instance. JSON in key-value pairs.
metadata	N	The VNF-specific metadata describing the VNF instance. JSON in key-value pairs.
extensions	N	The VNF-specific attributes that affect the lifecycle management of this VNF instance. JSON in key-value pairs.
scalingByMoreThanOneStepSupported	N	true, if scaling is supported by more than one step; otherwise false.
fdn	N	The FDN of the VNF as in ENM/OSS-RC. If provided, VNF-LCM do not try to update ENM/OSS-RC topology even if `addVnfToOss` is set to true. **Note:** Do not provide this parameter, if VNF-LCM is deployed as value pack to Ericsson Orchestrator-Cloud Manager.
onboardedVnfPkgInfoId	N	Identifier of information held by the NFVO about the specific VNF package on which the VNF is based. If not provided, then vnfdid is saved as `onboardedVnfPkgInfoId`.

Note: If VNF-LCM is deployed as a value pack to Ericsson Orchestrator-Cloud Manager, the properties in the table VNF scaleAspectInfo properties, VNF updateOssTopology properties, and AXE Properties are not applicable.

Table 48 VNF scaleAspectInfo Properties

Attribute	Required	Description
id	Y	Scale aspect id

Attribute	Required	Description
currentScaleLevel	Y	Current scale level of an instance.

Table 49 VNF updateOssTopology Properties

Attribute	Required		Description
	OSS-RC	ENM	
associatedSite	Y	Y	Mandatory. A site represents a geographical location for a Network Element (NE). The site under which the VNF is to be added in OSS-RC network resource model. In ENM, this parameter is not used.
communityString	N	N	Optional.Community string for SNMP. If not provided,default string `public` would be taken for OSS-RC and in ENM, default string `enm-public` would be taken. Not applicable for AXE based nodes and SNMP V3.
managedElementId	N	N	Managed element Id. If not provided, `vAppName` will be set to this parameter.
networkElementVersion	Y	N	Mandatory for OSSRC. The Version of the Network Element. For example, 15A, 15B, 16A, R15A. In ENM, this parameter refers to: ossModelIdentity like 16A-CP02 for non-AXE nodes and release for AXE based nodes. The attribute becomes mandatory

Attribute	Required		Description
			depending on the `networkElementType`.
collectionName	N	N	Applicable only for ENM. If `collectionName` is provided, the node is added to the given collection in ENM. Collection with the given collection name must be created in ENM before the use case is run, otherwise the node addition fails.
managedElementSrcType	Y	N	Mandatory. The `Src` type of network element. For example, WPP etc. In ENM, this parameter is not used. In OSSRC, for vMSC nodes, provide**`VIRTUAL`** and for vMSC(n+1) and vHLR nodes, provide **`VirtualHybrid`**. For vCUDB node, provide **`LINUX`**.
managedElementTypes	Y	N	Optional. Managed element type. In ENM, this parameter is not used.
netConfPort	N	Y	NetConf port number. In case OSS-RC, if not provided default port number `830` would be taken. In case of ENM, for `vEPG` it would be `830`, and for `vMME` it would be 22.
nodeIpAddress	Y	Y	Mandatory. The O&M IP address of the Network Element.

Attribute	Required		Description
nodePassword	Y	Y	Password for logging in to the VNF.
nodeUsername	Y	Y	Username for logging in to the VNF
snmpPort	N	N	Optional. SNMP port. Not applicable for AXE based nodes. If port is not provided, then the default port 161 will be set
subNetworks	N	N	Optional. SubNetwork is used for logical grouping of nodes within OSSRC and ENM. In OSS-RC, only a single child SubNetwork is allowed, apart from the Root (ONRM_ROOT_MO) SubNetwork. For example, if a SubNetwork called COMS to be added, then the value specified must contain SubNetwork=COMS. The unique identifier called FDN is SubNetwork=ONRM_ROOT_MO,SubNetwork=COMS,MeContext=UM1205,ManagedElement=UM1205. In ENM, it can contain one or more child SubNetwork. For example, if only one subNetwork, then the format must be "SubNetwork=CL" and if more than one child SubNetwork, then the format must be "SubNetwork=CL,SubNetwork=COMS". That

Attribute	Required		Description
			is, SubNetwork has to be separated by ",".
snmpVersion	N	N	NMP Version. Not applicable for AXE based nodes. In OSS-RC if version is not provided, then the default version V2c is set. In ENM for SNMP V3, it is mandatory to provide as "SNMP_V3".
snmpSecurityLevel	N	N	Mandatory and applicable only in case of SNMP V3. Not applicable for AXE based nodes. It is the permitted level of security within a security model. OSS-RC and ENM supports 3 security levels: AUTH_PRIV, AUTH_NO_PRIV, NO_AUTH_NO_PRIV. Security level "NO_AUTH_PRIV" is not supported.
snmpSecurityName	N	N	Mandatory and applicable only in case of SNMP V3. The security name for SNMP V3. Not applicable for AXE based nodes. In case of OSS-RC it should be same as networkElementUsername used for logging in to the VNF, if node security level is other than NO_AUTH_NO_PRIV.
snmpAuthProtocol	N	N	Optional.

Attribute	Required		Description
			Authentication protocol used to authorize the identity of user. Not applicable for AXE based nodes. If not provided, will be defaulted to 'MD5'. Applicable if SNMP V3 is used and the security level is either AUTH_NO_PRIV or AUTH_PRIV. In OSS-RC, this parameter refers to Authentication Method. Sample values: for OSSRC : 'SHA', 'MD5'. for ENM : 'MD5', 'SHA1'.
snmpPrivProtocol	N	N	Optional. Privacy Protocol used for encryption of SNMP v3 messages to ensure confidentiality of p. Not applicable for AXE based nodes. If not provided, will be defaulted to 'AES-128' in case of OSSRC and 'AES128' for ENM. Applicable if SNMP V3 is used and the security level is AUTH_PRIV. In OSS-RC, this parameter refers to Encryption Method. Sample values for OSSRC : 'AES-128', 'CBC-DES'. for ENM : 'AES128', 'DES'.
snmpAuthPassword	N	Y	Mandatory only for ENM when SNMP V3 is used and the security level is AUTH_NO_PRIV or

Attribute	Required		Description
			AUTH_PRIV. Password set for authentication with the provided Authentication Protocol for secure authorization.
snmpPrivacyPassword	N	Y	Mandatory only for ENM when SNMP V3 is used and the security level is AUTH_PRIV. Password set for authentication with the provided Privacy Protocol for secure authorization.
pmFunction	N	N	Optional in case of ENM. PmFunction is used to switch on and off PMIC Mediation flows for each Network Element. If PmFunction is on, all file collection, subscription activation, subscription deactivation, scanner polling and scanner master functionality is available for the Network Element. Default value is **`false`**.
cmNodeHeartbeatSupervision	N	N	Optional. This attribute is used only for ENM. Enabling CM Supervision copies the current status of the configuration data of the node into ENM and enable CM Notifications. Enabling CM Notifications ensures that the node notifies ENM of any configuration data

Attribute	Required		Description
			changes. Default value is **true**.
fmAlarmSupervision	N	N	Optional. Used only in case of ENM. FmAlarmSupervision is used to enable or disable alarms in case of any failures. Default value is **true**.
disableLdapUser	N	N	Optional. Whether to enable or disable node remote authentication with ENM COM-AA. If the parameter is not provided, the default is enable. Applicable only for ENM.
transportProtocol	N	N	Optional. Applicable only for ENM. If transportProtocol is not provided, then the default value SSH is set. Possible values are **SSH** and **TLS**.
vnfmId	N	N	Optional. VirtualNetworkFunctionManager ID applicable only for ENM. If vnfmId is not provided, then the configuration parameter vnfmId is read.
timeZone	N	N	Optional. Applicable only for AXE based nodes, vAFG and vEIR-FE nodes in ENM. If timeZone is not provided, then the timeZone of the system is set.
axe	N	N	Mandatory for AXE based nodes. AXE input JSON array, which contains inputs mention in table 41.1 AXE properties. This is

Attribute	Required		Description
			an array, may contain 1 or more elements.

Table 50 AXE Properties

Attribute	Required		Description
	OSS-RC	ENM	
axeIoApplications	Y	N	IO applications required for Network Element. Multiple comma separated values can be provided. For AXE nodes, it is mandatory to provide "APIO" as first application. For example: APIO, FOS, STS
axeNodeClusterIP	Y	Y	The Cluster IP Address of the Network Element.
axeNodeInterfaceAIp	Y	Y	IP address for the active side A of the Network Element
axeNodeInterfaceBIp	Y	Y	IP address for the active side B of the Network Element.
isManagerIpAddress	N	Y	Mandatory for IS-based AXE nodes. IP address for the SIS interface manager.

Note: For AXE nodes, all attributes are mandatory.

12.7.8.12.2 Template for updating VNF Scale Information

This section describes the template file - `template_scale_update.json`. This file is used to update the scale information of VNF in VNF-LCM. The template file is stored at `/ericsson/vnflcm/data` folder.

```
{
    "vnfInstanceName": "",
    "scalingByMoreThanOneStepSupported": "true",
```

```
    "scalingInfo": [{
            "id": "processing",
            "name": "processing",
            "description": "",
            "maxScaleLevel": 4,
            "currentScaleLevel": 1
        },
        {
            "id": "database",
            "name": "database",
            "description": "",
            "maxScaleLevel": 15,
            "currentScaleLevel": 1
        }
    ]
}
```

File Structure

Table 51 Update VNF Scale Info Properties

Attribute	Required	Description
vnfInstanceName	Y	VNF instance name
scalingByMoreThanOneStepSupported	N	true, if scaling is supported by more than one step; otherwise false
scalingInfo (inline)		
id	Y	scale aspect id
name	N	name of scale aspect.
description	N	description of scale aspect.
maxScaleLevel	Y	maximum level up to which scale is allowed.
currentScaleLevel	Y	current scale level of an instance.

12.7.9 Manage NFVO Configurations in VNF-LCM Framework

This utility is a replacement of existing nfvoconfig.json file. NFVO configuration is used to connect with NFVO and get the package information, send notification and get the grant for different use cases like instantiation, scale, heal and termination. In order to register NFVO with VNFLCM, the json file needs to be added to vnflcm through vnflcm admin cli. The json file contains 2 flags, isNotificationSupported and isGrantSupported. The values for attributes isNotificationSupported and isGrantSupported given in the file will depend whether grant and notifications are supported by NFVO.

12.7.9.1 Add New NFVO Configuration in VNF-LCM Framework

It is possible to add a second NFVO to VNF-LCM, however only one NFVO will be in use at a time. If request comes to create a nfvo with flag 'Y' for 'nfvoInUse' and an NFVO already exists with 'Y' flag, the existing flag will be set to 'N', and a new NFVO will be created with flag 'Y'. Any operations once started with a given NFVO will continue using that NFVO only until it is completed.

Prerequisites

- Root and cloud-user access to the vnflaf-services VM.
- Command console is open.
- Jboss/Server must be running.
- Basic understanding of JSON format.
- A back up of the `/ericsson/vnflcm/data/template_nfvoconfig.json` taken before executing this procedure.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Edit the template_nfvoconfig.json file to update its properties present under `/ericsson/vnflcm/data/`.

   ```
   [root@vnflaf-servcies tmp] vi template_nfvoconfig.json
   ```

   Note: Provide the complete file path if the file is not present in the same directory

   See template_nfvoconfig.json File on page 207 to learn more about this file.
3. Add new NFVO Configurations to VNF-LCM framework:

   ```
   root@vnflaf-services tmp] vnflcm nfvo add --file /tmp/template_nfvoconfig.js
   on
   ```

   Example:

   ```
   [root@vnflaf-services tmp]# vnflcm nfvo add --file /tmp/template_nfvoconfig.
   json
   Validations passed
   Restarting JBOSS...
   Restarting jboss (via systemctl): [ OK ]
   NFVO | Result
   --------------------------------+---------------------------------
   http://nfvohost:8080/ecm_service | NFVO addition successful
   ```

Note: Json file having been added, will be deleted post successful operation.

12.7.9.2 Update Existing NFVO Configuration in VNF-LCM Framework

Update information of an already-added NFVO in VNF-LCM to allow the workflow operations to run. If a request comes to update an nfvo with flag 'Y' for 'nfvoInUse', and an nfvo already exists with 'Y' flag, the existing flag will be set to 'N', and the new nfvo will be updated with flag 'Y'.

Prerequisites

- Root and cloud-user access to the vnflaf-services VM.
- Command console is open.
- Jboss/Server must be running.
- Basic understanding of JSON format.
- NFVO Configuration to be updated is already added.
- A back up of the `/ericsson/vnflcm/data/template_nfvoconfig.json` taken before running this procedure.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Edit the `template_nfvoconfig.json` file to update its properties present under `/ericsson/vnflcm/data/`.

```
[root@vnflaf-servcies tmp] vi template_nfvoconfig.json
```

Note: Provide the complete file path if the file is not present in the same directory.

See template_nfvoconfig.json File on page 207 to learn more about this file.

3. Update existing NFVO Configurations in VNF-LCM framework:

```
root@vnflaf-services tmp] vnflcm nfvo update --file /tmp/template_nfvoconfig →
.json
```

Example:

```
[root@Shashwat-vnflaf-services tmp]# vnflcm nfvo update --file /tmp/template →
_nfvoconfig.json
Validations passed
index | baseUrl | hostName | hostIpAddress | userName | subscriptionId
```

```
------+-----------------------------------+-----------+---------------+-----
-----+---------------
1 | http://nfvohost2:5000/ecm_service | nfvohost2 | 10.1.11.12 | vnflcm | 12
3000

Updating NFVO Configurations...
NFVO | Result
---------------------------------+----------------------------------------
---------
http://nfvohost2:5000/ecm_service | NFVO Configurations updated successfully
```

If more than one NFVOs with the same baseUrl are added, user will get a prompt to select the nfvo to be updated.

```
[root@Shashwat-vnflaf-services tmp]# vnflcm nfvo update --file /tmp/template
_nfvoconfig.json
Validations passed
index | baseUrl | hostName | hostIpAddress | userName | subscriptionId
------+-----------------------------------+----------+---------------+--------
--+---------------
1 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23 | vnflcm | 1234
56
2 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23 | vnflaf | 1357
90
Enter index of NFVO to perform the required operation: (-1 to cancel operati
on)
2

Updating NFVO Configurations...
NFVO | Result
-------------------------------+----------------------------------------
-------
http://nfvohost:8080/ecm_service | NFVO Configurations updated successfully
```

Note: The json file being updated is deleted after successful operation.

12.7.9.3 List Existing NFVO Configurations

List all NFVO Configurations in a framework.

Prerequisites

- Root and cloud-user access to the vnflaf-services VM.
- Command console is open.
- Jboss/Server must be running.
- NFVO Configuration present in VNF-LCM.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Run list command

```
[root@vnflaf-services ~]# vnflcm nfvo list [--baseurl BASEURL] [--nfvoinuse]
```

Example:

List all NFVO Configurations:

```
[root@vnflaf-services-0 ~]# vnflcm nfvo list
index | baseUrl                            | hostName | hostIpAddress | userN →
ame | subscriptionId
------+------------------------------------+----------+---------------+------ →
----+--------------
    1 | http://nfvohost:8080/ecm_service  | hostname | 10.1.11.23    | vnflc →
m   | 123456
    2 | http://nfvohost:8080/ecm_service  | hostname | 10.1.11.23    | vnfla →
f   | 135790
    3 | http://nfvohost2:5000/ecm_service | hostname | 10.1.11.12    | vnflc →
m   | 123000
```

List all NFVO Configurations using baseUrl

```
[root@-vnflaf-services-0 ~]# vnflcm nfvo list --baseurl http://nfvohost:8080 →
/ecm_service
index | baseUrl                           | hostName | hostIpAddress | userNa →
me | subscriptionId
------+-----------------------------------+----------+---------------+------- →
---+--------------
    1 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23    | vnflcm →
   | 123456
    2 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23    | vnflaf →
   | 135790
```

List NFVO in use:

```
[root@vnflaf-services-0 ~]# vnflcm nfvo list --nfvoinuse
index | baseUrl                            | hostName | hostIpAddress | userN →
ame | subscriptionId
------+------------------------------------+----------+---------------+------ →
----+--------------
    1 | http://nfvohost2:5000/ecm_service | hostname | 10.1.11.12    | vnflc →
m   | 123000
```

12.7.9.4 Delete NFVO Configuration from VNF-LCM Framework

Delete the NFVO Configuration when it is no longer required. If any instance is in a running state and has not completed, the user will not be allowed to delete that NFVO, if it is in use by that running instance. If any instance has completed (say Instantiation), NFVO will be deleted, but post that it will be treated as Small Stack case.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Command console is open.

— NFVO Configuration to be deleted exists in VNF-LCM.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Delete the relevant NFVO:

```
[root@vnflaf-services ~]# vnflcm nfvo delete --baseurl http://nfvohost2:5000 →
/ecm_service
```

Example:

```
[root@vnflaf-services ~]# vnflcm nfvo delete --baseurl http://nfvohost2:5000 →
/ecm_service
index | baseUrl                            | hostName  | hostIpAddress | user →
Name | subscriptionId
------+------------------------------------+-----------+---------------+----- →
-----+---------------
    1 | http://nfvohost2:5000/ecm_service | nfvohost2 | 10.1.11.12    | vnfl →
cm   | 123000
NFVO deleted successfully
```

If more than one NFVOs with the same baseUrl are added, user will get a prompt to select the nfvo to be deleted.

```
[root@vnflaf-services ~]# vnflcm nfvo delete --baseurl http://nfvohost:8080/ →
ecm_service
index | baseUrl                           | hostName | hostIpAddress | userNam →
e | subscriptionId
------+-----------------------------------+----------+---------------+-------- →
--+---------------
    1 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23    | vnflcm →
   | 123456
    2 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23    | vnflaf →
   | 135790
Enter index of NFVO to perform the required operation: (-1 to cancel operati →
on)
2
NFVO deleted successfully
```

12.7.9.5 Show Detailed Configuration of an Added NFVO

Show the complete configuration of an NFVO already added to VNF-LCM framework.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Command console is open.

— Jboss/Server must be running.

— NFVO Configuration present in VNF-LCM.

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Run the show command:

```
[root@vnflaf-services-0 ~]# vnflcm nfvo show --baseurl http://nfvohost:8080/ecm_service
```

Example (when more than one NFVOs are added with the provided baseUrl):

```
[root@vnflaf-services-0 ~]# vnflcm nfvo show --baseurl http://nfvohost:8080/ecm_service
index | baseUrl                          | hostName | hostIpAddress | userName | subscriptionId
------+----------------------------------+----------+---------------+----------+---------------
    1 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23    | vnflcm   | 123456
    2 | http://nfvohost:8080/ecm_service | hostname | 10.1.11.23    | vnflaf   | 135790
Enter index of NFVO to perform the required operation: (-1 to cancel operation)
1

Detailed NFVO Configuration for selected NFVO:
+--------------------------------+--------------------------------------+
| Name                           | Value                                |
+--------------------------------+--------------------------------------+
| id                             | 74528a60-bd5b-11e8-a103-fa163e34c4de |
| baseUrl                        | http://nfvohost:8080/ecm_service     |
| hostName                       | hostname                             |
| hostIpAddress                  | 10.1.11.23                           |
| userName                       | vnflcm                               |
| authType                       | Basic                                |
| isGrantSupported               | True                                 |
| isNotificationSupported        | True                                 |
| subscriptionId                 | 123456                               |
| nfvoType                       | ECM                                  |
| nfvoInUse                      | N                                    |
| orVnfmVersion                  | SOL231                               |
| notificationAckRequired        | YES                                  |
| staticAuthenticationTokenName  |                                      |
| staticAuthenticationTokenValue |                                      |
| nfvoEndPoints                  | Refer below table 'nfvoEndPoints':   |
| nfvoProperties                 | Refer below table 'nfvoProperties':  |
| tenancyDetails                 | Refer below table 'tenancyDetails':  |
+--------------------------------+--------------------------------------+

nfvoEndPoints:
+--------------------------+----------------------------------------+
| endPointName             | endPointUrl                            |
+--------------------------+----------------------------------------+
| grantUrl                 | /grant/v1/grants                       |
| authUrl                  | /tokens                                |
| lifecycleNotificationUrl | /VnfLcmOperationOccurrenceNotification |
| createNotificationUrl    | /vnf/v1/vnf_instances/creation         |
| deleteNotificationUrl    | /vnf/v1/vnf_instances/deletion         |
| packageManagementUrl     |                                        |
+--------------------------+----------------------------------------+

nfvoProperties:
+--------------------------------+-------------------------------------------------+
| propKey                        | propValue                                       |
+--------------------------------+-------------------------------------------------+
| nfvoSupportedNotificationTypes | CREATE, DELETE, START, PROCESSING, ROLLED_BACK, COMPLETED, FAILED, FAILED_TEMP, ROLLING_BACK|
+--------------------------------+-------------------------------------------------+
```

```
tenancyDetails:
+----------+--------------+--------------+------------------------------- →
-+
| tenantId | tenantName    | defaultTenant | vdcDetails                    →
 |
+----------+--------------+--------------+------------------------------- →
-+
| Tenant-1 | defaultTenant | True          | Refer below table 'vdcDetails': →
 |
+----------+--------------+--------------+------------------------------- →
-+

vdcDetails:
+----------+-------+------------+------------+
| tenantId | vdcId | vdcName    | defaultVdc |
+----------+-------+------------+------------+
| Tenant-1 | vdc-1 | defaultVdc | True       |
+----------+-------+------------+------------+
```

12.7.9.6 template_nfvoconfig.json File

> Description of configuration parameters in `template_nfvoconfig.json` file.

This file is used to add a new or update an existing NFVO Configuration in VNF-LCM framework.

Note: Although **tenancyDetails** is an array, but only single tenant and single vdc should be added. The array is kept to support future enhancements to support multiple tenants.

```
{
    "baseUrl": "",
    "hostName": "",
    "hostIpAddress": "",
    "userName": "",
    "password": "",
    "enmHostName": "",
    "authType": "OAUTH2_CLIENT_CREDENTIALS",
    "subscriptionId": "",
    "isGrantSupported": "true",
    "isNotificationSupported": "true",
    "nfvoType": "ECM",
    "nfvoInUse": "Y",
    "orVnfmVersion": "SOL241",
    "clientId": "",
    "clientSecret": "",
    "notificationAckRequired": "YES",
    "tenancyDetails": [{
        "tenantId": "",
        "tenantName": "",
        "defaultTenant": "true",
        "vdcDetails": [{
            "id": "",
            "name": "",
            "defaultVdc": "true"
        }]
    }],
    "nfvoEndPoints": [{
            "endPointName": "grantUrl",
            "endPointUrl": "/grant/v1/grants"
        },
        {
            "endPointName": "authUrl",
            "endPointUrl": "/oauth2/access_token"
        },
        {
```

```
            "endPointName": "lifecycleNotificationUrl",
            "endPointUrl": "/VnfLcmOperationOccurrenceNotification"
        },
        {
            "endPointName": "createNotificationUrl",
            "endPointUrl": "/vnf/v1/vnf_instances/creation"
        },
        {
            "endPointName": "deleteNotificationUrl",
            "endPointUrl": "/vnf/v1/vnf_instances/deletion"
        },
        {
            "endPointName": "packageManagementUrl",
            "endPointUrl": "/vnfpkgm/v1/vnf_packages"
        },
        {
            "endPointName": "queryVdcUrl",
            "endPointUrl": "/vdcs"
        }
    ],
    "nfvoProperties": [{
            "propKey": "nfvoSupportedNotificationTypes",
            "propValue": "CREATE, DELETE, START, PROCESSING, ROLLED_BACK, COMPLE →
TED, FAILED"
        },
        {
            "propKey": "isfallbackBestEffortSupported",
            "propValue": "Yes"
        },
        {
            "propKey": "staticAuthenticationTokenName",
            "propValue": ""
        },
        {
            "propKey": "staticAuthenticationTokenValue",
            "propValue": ""
        }
    ]
}
```

File Structure Explained:

Table 52 NFVO Config. Properties

Attribute	Required	Description
authType	N	Defines the type of Authentication / Authorization which the API consumer is willing to accept when receiving a notification. Permitted values: — `BASIC`: For Basic authentication. — `OAUTH2_CLIENT_CREDENTIALS`: For OAuth 2.0 Bearer token based authentication. If value not provided, then the default `OAUTH2_CLIENT_CREDENTIALS` will be set.
baseUrl	Y	baseUrl of an NFVO to be used. Format for the baseURl is *<ECM-GUI-URL>*`/ecm_service`.
clientId	N	Required only if 'authType' is 'OAUTH2_CLIENT_CREDENTIALS'. The client identifier issued to the client by the authorization server during the registration process.

Attribute	Required	Description
clientSecret	N	Required only if 'authType' is 'OAUTH2_CLIENT_CREDENTIALS'. The client secret is a secret known only to the application and the authorization server.
enmHostName	N	Httpd Hostname for ENM/OSS.
hostIpAddress	Y	hostIpAddress of an NFVO to be used.
hostName	Y	hostName of an NFVO to be used.
isGrantSupported	N	Determines if grant is supported on NFVO. Allowed values : ”true”/”false”. If value not provided, then the default "false" will be set.
isNotificationSupported	N	Determines if notification is supported on NFVO. Allowed values : ”true”/”false”. If value not provided, then the default "false" will be set.
nfvoEndPoints	Y	End points of an NFVO.
nfvoInUse*	Y	Determines if the NFVO is to be used for different use-cases like Instantiation, Scale, Heal, Termination.
nfvoProperties	Y	Additional properties of an NFVO.
nfvoType	Y	Type of NFVO
notificationAckRequired	N	Determines if the Notification Acknowledge required. Valid values are YES and NO. If not provided, the default "YES" will be set.
orVnfmVersion	N	Determines if the orVnfm version supported is Sol003v241 or SOL003v231. Valid values are SOL241 and SOL231. If not provided, the default "SOL231" will be set.
password	Y	Password for nfvo authorization.
subscriptionId	Y	Id generated when a VNFM registered in NFVO.
tenancyDetails	Y	Tenant details on which vnf package deployed. **Note:** Only single tenant with single vdc detail is supported. Multi-tenancy support is not provided. Please provide only single entry.
userName	Y	Username for Nfvo authorization.

Table 53 Tenancy Details Properties

Attribute	Required	Description
tenantId	Y	Tenantid in nfvo (required if nfvo is ecm) for authorization.
tenantName	N	Name of Tenant in NFVO.

Attribute	Required	Description
defaultTenant	N	Set it to True if this tenant is used as default Tenant in cases where workflow are executed in non-interactive mode. If value not provided, then the default "true" will be set.
vdcDetails	N	Virtual Deployment Center info on which vnf package is deployed. **Note:** Only single tenant with single vdc detail is supported. Multi-tenancy support is not provided. Please provide only single entry.

Table 54 VDC Details Properties

Attribute	Required	Description
id	Y	dcid in nfvo (required if nfvo is ecm) for create Vnf Identifier. Mandatory to provide if vdcDetails is to be considered.
name	N	Name of VDC in NFVO
defaultVdc	N	Set it to True if this vdcis used as default vdc in cases where workflow are executed in non-interactive mode. If value not provided, then the default "true" will be set.

Table 55 nfvoEndPoints Properties

endPointName	endPointUrl	Required	Description
grantUrl	`/grant/v1/grants`	N	NFVO grant authorization, which allows the VNFM to obtain the NFVO permission and configuration parameters for the VNF lifecycle operation. This endpoint is applicable when `isNotificationSupported` is set to **`True`** in Table 52. `endPointUrl` is according to SOL003.
authUrl	`/oauth2/access_token`	Y	Endpoint for NFVO authorization. If the authorization server is different to NFVO baseUrl, provide the fully qualified URI of the authorization server.
lifecycleNotificationUrl	`/VnfLcmOperationOccurrenceNotification`	N	NFVO notification endpoint for VNF Life Cycle Management operation occurrence. This endpoint is applicable when `isNotificationSupported` is set to **`True`** in Table

endPointName	endPointUrl	Required	Description
			52. endPointUrl is according to SOL003.
createNotificationUrl	/vnf/v1/vnf_instances/creation	N	NFVO notification endpoint for VNF identifier creation notification.This endpoint is applicable when isNotificationSupported is set to **True** in Table 52. endPointUrl is according to SOL003.
deleteNotificationUrl	/vnf/v1/vnf_instances/deletion	N	NFVO notification endpoint for VNF identifier deletion notification.This endpoint is applicable when isNotificationSupported is set to **True** in Table 52. endPointUrl is according to SOL003.
packageManagementUrl	/vnfpkgm/v1/vnf_packages	N	Endpoint of Automatic package management to onboard the VNF package. This is applicable when VNF package is required to be downloaded from the NFVO automatically. endPointUrl is according to SOL003.
queryVdcUrl	/vdcs	N	Endpoint for TOSCA instantiation where user can select the VDC in VNFLCM GUI. Workflows are not using this input in the GUI.
Note: 'endPointName' must be same as mentioned in table			

Table 56 nfvoProperties

Attribute	Required	Description
propKey	N	Key for any static value which to be exposed by NFVO like authentication token from VDC
propValue	N	static value exposed by NFVO for a given propKey like authentication token from VDC
nfvoSupportedNotificationTypes	N	The notification types supported by a Nfvo. By default all ECM supported operation states will be present. **Note:** ROLLING_BACK is not supported for ECM.
isfallbackBestEffortSupported	N	Default value is yes. When set to yes, the fallbackBestEffort variable is added to placement constraint attribute. When set to no, the fallbackBestEffort variable is not added to placement constraint attribute of grant request. **Note:**

Attribute	Required	Description
		For SOL003v2.6.1 onwards, `fallbackBestEffort` variable is supported in placement constraint.
`staticAuthenticationTokenName`	N	This property is configured only when static token is shared by NFVO. VNF-LCM should use that token for communication with NFVO.
`staticAuthenticationTokenValue`	N	This property is configured only when static token is shared by NFVO. VNF-LCM should use that token for communication with NFVO.
Note: It is mandatory to provide "nfvoSupportedNotificationTypes" as 'propKey' and the corresponding values as 'propValue'. If user does not provide this, then default entry as below will be set considering ECM as NFVO: {"propKey": "nfvoSupportedNotificationTypes","propValue": "CREATE, DELETE, START, PROCESSING, ROLLED_BACK, COMPLETED, FAILED, FAILED_TEMP, ROLLING_BACK"}		

Note: * Or-Vnfm Interfaces requires ENM Authentication for NFVO transactions.

12.7.10 Manage EM Configurations in VNF-LCM Framework

This utility allows the administrator to add `Element Manager (EM) Configuration`. EM configuration is used to connect with EM, and send notification for different use cases, as per SOL002 v2.4.1 standards. In order to register EM with VNF-LCM, the `json` file must be added to `vnflcm` through `Vnflcm Admin CLI`. The `json` file contains one flag, `isNotificationSupported`. The value for attribute `isNotificationSupported` given in the file depends on whether notifications are supported by EM.

12.7.10.1 Add New EM Configuration

> VNF-LCM allows having more than one EM in use `(nfvoInUse=Y)` at a time. For example, on request, VNF-LCM creates a new EM with **Y** flag for `nfvoInUser` even if there is an existing EM with flag **Y**, resulting two EMs in use.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— Command console is open.

— Jboss or Server must be running.

— Basic understanding of JSON format.

— A back up of the `/ericsson/vnflcm/data/template_em_nfvoconfig.json` taken before starting this procedure.

Expected Result

EM information is added.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, and switch to root user.
2. Edit the `template_em_nfvoconfig.json` file to update its properties present under `/ericsson/vnflcm/data/`.

   ```
   [root@vnflaf-services-0 ~]# vi template_em_nfvoconfig.json
   ```

   Provide complete file path where the file is not in the same directory.

   For more information, see template_em_nfvoconfig.json File on page 219.
3. Add new EM Configurations:

   ```
   [root@vnflaf-services-0 ~]# vnflcm em add --file /tmp/template_em_nfvoconfig
   .json
   ```

   Example

   ```
   [root@vnflaf-services-0 ~]# vnflcm em add --file /tmp/template_em_nfvoconfig
   .json
   Validations passed
   NFVO | Result
   -------------------------------+---------------------------------
   http://localhost:58585          | EM addition successful
   ```

   Note: `Json` file is deleted after successful operation.

12.7.10.2 Update Existing EM Configuration

> Update EM information in a VNF-LCM to allow the workflow operations to run. If request comes to update a EM with flag **Y** for `nfvoInUse`, and another EM already exists with **Y** flag then VNF-LCM allows both EMs to be in use. Provide the `baseUrl` of the EM to be updated in the `json` file.

Prerequisites

- Root and cloud-user access to the `vnflaf-services` VM.
- Command console is open.
- Jboss or Server must be running.
- Basic understanding of JSON format.
- EM Configuration to be updated is already present.

— A backup of the `/ericsson/vnflcm/data/template_em_nfvoconfig.json` taken before running this procedure.

Expected Result

EM information of an EM is updated in VNF-LCM.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, and switch to root user.
2. Edit the `template_em_nfvoconfig.json` file to update its properties present under `/ericsson/vnflcm/data/`.

```
[root@vnflaf-services-0 ~]# vi template_em_nfvoconfig.json
```

Provide complete file path where the file is not in the same directory.

For more information, see template_em_nfvoconfig.json File on page 219.

3. Update existing EM configurations:

```
[root@vnflaf-services-0 ~]# vnflcm em update --file /tmp/template_em_nfvocon →
fig.json
```

Example

```
[root@vnflaf-services-0 ~]# vnflcm em update --file /tmp/template_em_nfvocon →
fig.json
Validations passed
index | baseUrl | hostName | hostIpAddress | userName | subscriptionId
------+-----------------------------------+-----------+---------------+-----  →
-----+---------------
1 | http://localhost:58585 | localhost| 127.0.0.1 | vnflcm | 123000
hostName, hostIpAddress & tenancyDetails are not allowed to update
Updating EM Configurations...
NFVO | Result
---------------------------------+----------------------------------------  →
---------
http://localhost:58585           | EM Configurations updated successfully
```

If more than one EM with the same `baseUrl` are added, user is prompted to select the EM to update.

```
[root@vnflaf-services-0 ~]# vi em.json
[root@vnflaf-services-0 ~]# vnflcm em update --file em.json
Validations passed
+-------+-----------------------------------+-----------+---------------+-  →
---------+----------------+
| index |              baseUrl              |  hostName | hostIpAddress |   →
userName | subscriptionId |
+-------+-----------------------------------+-----------+---------------+-  →
---------+----------------+
|   1   |       http://127.0.0.1:58585      | localhost |   127.0.0.1   |   →
  test   |      1111      |
|   2   | http://enmHost:8080/sol002Interface |  enmHost  |  10.190.11.25 |   →
vnfuser  |     987654     |
+-------+-----------------------------------+-----------+---------------+-  →
---------+----------------+
Enter index of {} to perform the required operation: (-1 to cancel operation →
```

```
)
1
Updating EM Configurations...
+------------------------+-----------------------------------------+
|          NFVO          |                 Result                  |
+------------------------+-----------------------------------------+
| http://127.0.0.1:58585 | EM Configurations updated successfully |
+------------------------+-----------------------------------------+
```

Note: `Json` file is deleted after successful operation.

12.7.10.3 List Existing EM Configurations

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— Command console is open.

— Jboss or Server must be running.

— EM Configurations are present.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, and switch to root user.
2. Run list command:

```
[root@vnflaf-services-0 ~]# vnflcm em list [--baseurl BASEURL] [--eminuse]
```

Example

List all EM configurations:

```
[root@vnflaf-services-0 ~]# vnflcm em list
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
| index |         baseUrl         |  hostName | hostIpAddress | userName | s →
ubscriptionId |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
|   1   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |   →
   12345      |
|   2   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |   →
   12345      |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
```

List all EM Configurations using baseUrl:

```
[root@vnflaf-services-0 ~]# vnflcm em list --baseurl https://localhost:58585
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
| index |         baseUrl         |  hostName | hostIpAddress | userName | s →
ubscriptionId |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
```

```
|   1   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |    →
   12345      |
|   2   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |    →
   12345      |
+-------+-------------------------+-----------+---------------+----------+--  →
--------------+
```

List EM in use:

```
[root@vnflaf-services-0 ~]# vnflcm em list --eminuse
+-------+-------------------------+-----------+---------------+----------+--  →
--------------+
| index |         baseUrl         |  hostName | hostIpAddress | userName | s  →
ubscriptionId |
+-------+-------------------------+-----------+---------------+----------+--  →
--------------+
|   1   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |    →
   12345      |
+-------+-------------------------+-----------+---------------+----------+--  →
--------------+
```

12.7.10.4 Delete EM Configuration

> Delete the EM Configuration when no longer required.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— Command console is open.

— EM Configuration to be deleted exists.

Expected Result

EM information of an EM is deleted from VNF-LCM.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, and switch to root user.
2. Fetch the `baseUrl`:

```
[root@vnflaf-services-0 ~]# vnflcm em list
id                                   | baseUrl                →
            | hostName  | subscriptionId
-------------------------------------+------------------------ →
------------+-----------+---------------
5013f815-b294-11e8-ae1f-fa163e1b0b18 | http://nfvohost:8080/e →
cm_service   | hostname  | 123456
986295ae-b29c-11e8-a8e9-fa163e1b0b18 | http://nfvohost2:5000/ →
ecm_service | nfvohost2 | 123000
```

```
879e5ec3-b29e-11e8-a8e9-fa163e1b0b18 | http://nfvohost:8080/e →
cm_service   | hostname  | 135790
```

3. Delete the relevant EM :

```
[root@vnflaf-services-0 ~]# vnflcm em delete --baseurl http://nfvohost2:5000 →
/ecm_service
```

Example

```
[root@vnflaf-services-0 ~]# vnflcm em delete --baseurl http://localhost:5858 →
5
index | baseUrl                           | hostName  | hostIpAddress | user →
Name | subscriptionId
------+-----------------------------------+-----------+---------------+----- →
-----+--------------
    1 | http://localhost:58585            | nfvohost2 | 10.1.11.12    | vnfl →
cm   | 123000
EM deleted successfully
```

If more than one EM with the same `baseUrl` is added, user is prompted to select the EM to delete.

```
[root@vnflaf-services-0 ~]# vnflcm em delete --baseurl https://localhost:585 →
85
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
| index |         baseUrl         |  hostName | hostIpAddress | userName | s →
ubscriptionId |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
|   1   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |   →
   12345      |
|   2   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |   →
   12345      |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
Enter index of {} to perform the required operation: (-1 to cancel operation →
)
2
EM deleted successfully
```

12.7.10.5 Show Detailed Configuration of an Added EM

> Show the complete configuration of an EM added to VNF-LCM.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— Command console is open.

— Jboss or Server must be running.

— EM Configuration is available.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, and switch to root user.
2. Fetch the `baseUrl`:

```
[root@vnflaf-services-0 ~]# vnflcm em list
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
| index |         baseUrl         |  hostName | hostIpAddress | userName | s →
ubscriptionId |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
|   1   | https://localhost:58585 | localhost |   127.0.0.1   | vnfuser  |   →
   12345      |
+-------+-------------------------+-----------+---------------+----------+-- →
--------------+
```

3. Show the detailed EM Configuration using the `baseUrl` fetched in the previous step:

```
[root@vnflaf-services-0 ~]# vnflcm em show --baseurl https://localhost:58585
```

Example

```
[root@vnflaf-services-0 ~]# vnflcm em show --baseurl  http://127.0.0.1:58585
+-------+-------------------------+-----------+---------------+----------+--- →
-------------+
| index |        baseUrl          |  hostName | hostIpAddress | userName | su →
bscriptionId |
+-------+-------------------------+-----------+---------------+----------+--- →
-------------+
|   1   | http://127.0.0.1:58585 | localhost |   127.0.0.1   | testxyz1 |     →
   1111      |
+-------+-------------------------+-----------+---------------+----------+--- →
-------------+

Detailed EM Configuration for selected EM:
+-------------------------+--------------------------------------+
| Name                    | Value                                |
+-------------------------+--------------------------------------+
| id                      | 46da5dfc-6418-11ea-9122-fa163e6d8413 |
| baseUrl                 | http://127.0.0.1:58585               |
| hostName                | localhost                            |
| hostIpAddress           | 127.0.0.1                            |
| userName                | testxyz1                             |
| authType                | EM_AUTH                              |
| isGrantSupported        | False                                |
| isNotificationSupported | True                                 |
| subscriptionId          | 1111                                 |
| nfvoType                | EM                                   |
| nfvoInUse               | Y                                    |
| orVnfmVersion           | SOL241                               |
| notificationAckRequired | YES                                  |
| tokenAttribute          | Credentials                          |
| tokenType               | None                                 |
| tokenPresence           | Body                                 |
| tokenHeader             | Authtoken                            |
| nfvoEndPoints           | Refer below table 'nfvoEndPoints'    |
| nfvoProperties          | Refer below table 'nfvoProperties'   |
| tenancyDetails          | Refer below table 'tenancyDetails'   |
+-------------------------+--------------------------------------+

nfvoEndPoints:
+--------------------------+-----------------------+
| endPointName             | endPointUrl           |
+--------------------------+-----------------------+
| authUrl                  | /login                |
| lifecycleNotificationUrl | /NotificationEndpoint |
| createNotificationUrl    | /NotificationEndpoint |
| deleteNotificationUrl    | /NotificationEndpoint |
```

```
+-------------------------+----------------------+

nfvoProperties:
+-------------------------------+-------------------------------------------- →
-------------------------------------------------+
| propKey                       | propValue                                   →
                                                 |
+-------------------------------+-------------------------------------------- →
-------------------------------------------------+
| nfvoSupportedNotificationTypes | CREATE, DELETE, START, PROCESSING, ROLLED →
_BACK, COMPLETED, FAILED, ROLLING_BACK, FAILED_TEMP |
+-------------------------------+-------------------------------------------- →
-------------------------------------------------+

tenancyDetails:
+----------+------------+---------------+---------------------------------+
| tenantId | tenantName | defaultTenant | vdcDetails                      |
+----------+------------+---------------+---------------------------------+
| tid666   |            | True          | Refer below table 'vdcDetails': |
+----------+------------+---------------+---------------------------------+

vdcDetails:
+----------+-------------+------+------------+
| tenantId | id          | name | defaultVdc |
+----------+-------------+------+------------+
| tid666   | uyu786jhklj |      | True       |
+----------+-------------+------+------------+
```

12.7.10.6 template_em_nfvoconfig.json File

> Description of configuration parameters in `template_em_nfvoconfig.json` file. This file is used to add or update an EM Configuration in VNF-LCM.

Note:

- Although `tenancyDetails` is an array, only single tenant and single vdc should be added. The array is kept to for future enhancements to support multiple tenants.
- `tenancyDetails` is optional parameter. If provided, validation is performed for it.

```
{
    "baseUrl": "",
    "hostName": "",
    "hostIpAddress": "",
    "userName": "",
    "password": "",
    "authType": "EM_AUTH",
    "subscriptionId": "",
    "isNotificationSupported": "true",
    "nfvoType": "EM",
    "nfvoInUse": "Y",
    "orVnfmVersion": "SOL241",
    "notificationAckRequired": "YES",
    "tenancyDetails": [{
        "tenantId": "",
        "tenantName": "",
        "defaultTenant": "true",
        "vdcDetails": [{
            "id": "",
            "name": "",
            "defaultVdc": "true"
        }]
    }],
    "nfvoEndPoints": [{
            "endPointName": "authUrl",
            "endPointUrl": "/login"
```

```
        },
        {
            "endPointName": "lifecycleNotificationUrl",
            "endPointUrl": "/VnfLcmOperationOccurrenceNotification"
        },
        {
            "endPointName": "createNotificationUrl",
            "endPointUrl": "/vnf/v1/vnf_instances/creation"
        },
        {
            "endPointName": "deleteNotificationUrl",
            "endPointUrl": "/vnf/v1/vnf_instances/deletion"
        }
    ],
    "nfvoProperties": [{
        "propKey": "nfvoSupportedNotificationTypes",
        "propValue": "CREATE, DELETE, START, PROCESSING, ROLLED_BACK, COMPLETED,
 FAILED"
    }]
}
```

12.7.10.7 File Structure Explained

Table 57 NFVO Config. properties

Attribute	Required	Description
baseUrl	Y	baseUrl of an EM to be used.
hostName	Y	hostName of an EM to be used.
hostIpAddress	Y	hostIpAddress of an EM to be used.
userName	Y	Username for EM authorization.
password	Y	Password for EM authorization.
enmHostName	N	Httpd Hostname for ENM/OSS.
authType	N	If value not provided, then the default "Basic" will be set.
isNotificationSupported	N	Determines if notification is supported on EM . Allowed values : “true”/”false”. If value not provided, then the default "false" will be set.
nfvoType	Y	Determines if the EM is to be used for different use cases like

Attribute	Required	Description
		Instantiation, Scale, Heal, Termination.
nfvoInUse	Y	Determines if the EM is to be used for different use cases like Instantiation, Scale, Heal, Termination.
tenancyDetails	N	Tenant details on which vnf package deployed.Optional. **Note:** Only single tenant with single vdc detail is supported. Multi-tenancy support is not provided. Please provide only single entry.
nfvoEndPoints	N	End points of an EM.
nfvoProperties	Y	Additional properties of an EM.

Table 58 Tenancy Details Properties (Optional)

Attribute	Required	Description
tenantId	Y	Tenantid in EM for authorization.
tenantName	N	Name of Tenant in EM.
defaultTenant	N	Set it to True if this tenant is used as default Tenant in cases where workflow are executed in non-interactive mode. If value not provided, then the default "true" will be set.
vdcDetails	N	Virtual Deployment Center info on which vnf package is deployed. **Note:**

Attribute	Required	Description
		Only single tenant with single vdc detail is supported. Multi-tenancy support is not provided. Please provide only single entry.

Table 59 VDC Details Properties (Optional)

Attribute	Required	Description
`id`	Y	vdcid in EM for create Vnf Identifier. Mandatory to provide if vdcDetails is to be considered.
`name`	N	Name of VDC in EM
`defaultVdc`	N	Set it to True if this vdc is used as default vdc in cases where workflow are executed in non-interactive mode. If value not provided, then the default "true" will be set.

Table 60 EM EndPoints Properties

Attribute	endPointUrl	Required	Description
`authUrl`	`/login`	Y	Endpoint for EM authorization, which must be login endpoint for EM.
`lifecycleNotificationUrl`	`/VnfLcmOperationOccurrenceNotification`	N	EM notification endpoint for VNF lifecycle management operation occurrence. This is applicable when `isNotificationSupported` is set to `True` in Table 57.

Attribute	endPointUrl	Required	Description
createNotificationUrl	/vnf/v1/vnf_instances/creation	N	EM notification endpoint for VNF identifier creation notification. This is applicable when isNotificationSupported is set to True in Table 57.
deleteNotificationUrl	/vnf/v1/vnf_instances/deletion	N	EM notification endpoint for VNF identifier deletion notification. This is applicable when isNotificationSupported is set to True in Table 57.
Note: 'endPointName' must be same as mentioned in table.			

Table 61 nfvoProperties

Attribute	Required	Description
nfvoSupportedNotificationTypes	Y	The notification types supported by a EM . By default all SOL002 supported operation states are present.
Note: propKey and propValue must be same as mentioned in table and template_nfvoconfig.json		

12.7.11 ENM Details to VNF-LCM

New CLIs in VNF-LCM tool to add, list, and update ENM admin detail.

Prerequisites:

— Root and cloud-user access to the vnflaf-services VM.

— Command console is open.

12.7.11.1 Add ENM Details to VNF-LCM Framework

This utility is provided to add ENM details to VNF-LCM Framework

Steps

1. Log on to vnflaf-services VM as cloud-user, then switch to root user.
2. Add ENM details to VNF-LCM Framework:

```
[root@vnflaf-services ~]# vnflcm enm add --enmHostName=<value> --enmUser=<va →
lue> --enmPassword=<value> --enmAuthRequired=<optional value>
```

Example

```
[root@vnflaf-services ~]# vnflcm enm add --enmHostName=ieatenm5325-1.athtem. →
eei.ericsson.se --enmUser=administrator --enmPassword=TestPassw0rd
--enmAuthRequired=Yes
Configuring certificates...
Restarting JBOSS...
Restarting jboss (via systemctl):           [ OK ]
ENM admin details added successfully.
```

Field Name	Optional/Mandatory	Description
`enmUser`	Mandatory	Provide valid user with SECURITY_ADMIN role.
`enmPassword`	Mandatory	Provide ENM password for respective ENM user.
`enmHostName`	Mandatory	Provide ENM host name.
`enmAuthRequired`	Optional	yes/no. This field used to enable or disable RBAC feature.

12.7.11.2 Update ENM Details to VNF-LCM Framework

This utility is provided to update ENM details on the VNF-LCM Framework

Steps

1. Log on to `vnflaf-services` VM as cloud-user, then switch to root user.
2. Update ENM details on the VNF-LCM Framework:

   Note: While updating, all fields are optional. Update of one, or only some fields is allowed.

```
[root@vnflaf-services ~]# vnflcm enm update --enmHostName=<value> --enmUser=
<value> --enmPassword=<value> --enmAuthRequired=<value>
```

Example

```
[root@vnflaf-services ~]# vnflcm enm update --enmHostName=ieatenm5325-1.atht
em.eei.ericsson.se --enmUser=administrator --enmPassword=TestPassw0rd
--enmAuthRequired=Yes
ENM admin details updated successfully
```

Field Name	Optional/Mandatory	Description
`enmUser`	Optional	Provide valid user with SECURITY_ADMIN role.
`enmPassword`	Optional	Provide ENM password for respective ENM user.
`enmHostName`	Optional	provide ENM host name.
`enmAuthRequired`	Optional	yes/no. This field used for RBAC feature is enable or disable.

12.7.11.3 List ENM details to VNF-LCM Framework

This utility is provided to list the ENM details from VNF-LCM Framework

Steps

1. Log on to `vnflaf-services` VM as cloud-user, then switch to root user.
2. List ENM details from VNF-LCM Framework:

```
[root@vnflaf-services ~]# vnflcm enm list
```

Example

```
[root@vnflaf-services ~]# vnflcm enm list
+-----------------+---------------+--------------------------------------+
| enmAuthRequired |    enmUser    |             enmHostName              |
+-----------------+---------------+--------------------------------------+
|       Yes       | administrator | ieatenm5325-1.athtem.eei.ericsson.se |
|                 |               |                                      |
+-----------------+---------------+--------------------------------------+
```

12.7.12 VNF-LCM Security Utility

The VNF-LCM Security Utility provides the ability to allow and remove access to the VNF-LCM eth0 (internal) and eth1 (external) interfaces via IPtable rules. The

`iptable` rules will be applied based on the service names, hostnames, or IP addresses provided in the input file.

Note: Prior to security hardening on external interface, consider the following :

- GUI Launch via browser: In order to launch the GUI, the IP address of the local system must be allowed on the external interface.
- Upgrade: Where security is enabled on external interface, and the upgrade is carried out using `autodeployer` tool, then it is mandatory to allow the hostname where upgrade job has to be carried out.

Prerequisites

- Root and cloud-user access to the `vnflaf-services` VM.
- VNF-LCM should be in healthy state.
- Command console is open.
- After initial deployment no rules exist, and traffic on all ports to and from all IPs are allowed.

12.7.12.1 Applying Security for VNF-LCM with ENM on Cloud

This procedure allows access towards VNF-LCM for the list of services, hostnames, or IPs provided on a specific interface.

On the external interface (`eth1`), the user is responsible for the creation of a file with the list of service names, hostnames, or IP addresses, followed by executing the procedure toward `eth1`.

Note: If service names or hostnames are used in this file, they must be resolvable externally by VNF-LCM. Otherwise IP addresses must be used. Once the security is applied to VNF-LCM `eth1`, only those specified IPs will be able to communicate towards VNF-LCM, including VNF-LCM UI and it's REST interface.

On the internal interface (eth0), the list of services, hostnames, or IPs that are required to communicate towards VNFLCM are provided in `enm_iptables_white_list.txt` file, stored under `/vnflcm-ext/enm`. Do not edit this file.

SSH port will always remain open.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, then switch to root user.

2. Allow access towards VNF-LCM from the services, hostnames, or IPs specified in the input file:

```
[root@vnflaf-services ~]# vnflcm security allowaccess [--interface <eth0/eth →
1> --port <PORT> --file <FILE>]
```

Note: a. **--file <FILE>**

i. If interface is internal, and file is not provided, default file `enm_iptables_white_list.txt` present under `/vnflcm-ext/enm/` will be used.

ii. If interface is external, file is a mandatory parameter.

iii. Only text files(.txt) are allowed. Entries in a file must be comma-separated IPs, services, or hostnames.

b. **--port <PORT>**

This is an optional parameter that allows for additional ports to be specified and added to the `iptables` rules. By default port 80 and 8080 are always allowed.

Examples:

Note: Failure messages as in the following examples are expected when IPv6 is not supported.

Allow Access to Services:

```
[root@vnflaf-services ~]# vnflcm security allowaccess --interface eth0 --por →
t 80 --file /vnflcm-ext/enm/enm_iptables_white_list.txt
This CLI will allow traffic on port 80 for IPs/services provided in file. Ar →
e you sure to run the CLI? (y/n)y
SUCCESS: Added ipv4, ACCEPT rule for cnom, port 80 on interface eth0
FAILED: ipv6, cnom is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for elasticsearch, port 80 on interface eth →
0
FAILED: ipv6, elasticsearch is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for httpd, port 80 on interface eth0
FAILED: ipv6, httpd is not resolvable. Provide valid service name/IP
Access allowed to services/IPs successfully
```

Allow Access to Services : Port is not Provided and File is Provided:

```
[root@vnflaf-services ~]# vnflcm security allowaccess --interface eth0 --fil →
e /vnflcm-ext/enm/enm_iptables_white_list.txt
This CLI will allow traffic on port 80 and 8080 for IPs/services provided in →
 file. Are you sure to run the CLI? (y/n)y
SUCCESS: Added ipv4, ACCEPT rule for cnom, port 80 on interface eth0
FAILED: ipv6, cnom is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for cnom, port 8080 on interface eth0
FAILED: ipv6, cnom is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for elasticsearch, port 80 on interface eth →
0
FAILED: ipv6, elasticsearch is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for elasticsearch, port 8080 on interface e →
th0
FAILED: ipv6, elasticsearch is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for httpd, port 80 on interface eth0
FAILED: ipv6, httpd is not resolvable. Provide valid service name/IP
```

```
SUCCESS: Added ipv4, ACCEPT rule for httpd, port 8080 on interface eth0
FAILED: ipv6, httpd is not resolvable. Provide valid service name/IP
Access allowed to services/IPs successfully
```

Allow Access to White_List Services : Port and File are not Provided:

```
[root@vnflaf-services ~]# vnflcm security allowaccess --interface eth0
File path not provided. CLI will take enm_iptables_white_list.txt file from  →
/vnflcm-ext/enm/ and only allow traffic on port 80 and 8080 for IPs/services →
 provided in file. Are you sure to run the CLI? (y/n)y
SUCCESS: Added ipv4, ACCEPT rule for cnom, port 80 on interface eth0
FAILED: ipv6, cnom is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for cnom, port 8080 on interface eth0
FAILED: ipv6, cnom is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for elasticsearch, port 80 on interface eth →
0
FAILED: ipv6, elasticsearch is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for elasticsearch, port 8080 on interface e →
th0
FAILED: ipv6, elasticsearch is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for httpd, port 80 on interface eth0
FAILED: ipv6, httpd is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for httpd, port 8080 on interface eth0
FAILED: ipv6, httpd is not resolvable. Provide valid service name/IP
Access allowed to services/IPs successfully
```

Allow Access to White_List Services : Port is Provided and File is not Provided:

```
[root@vnflaf-services ~]# vnflcm security allowaccess --interface eth0 --por →
t 8080
File path not provided. CLI will take enm_iptables_white_list.txt file from  →
/vnflcm-ext/enm/ and only allow traffic on port 8080 for IPs/services provid →
ed in file. Are you sure to run the CLI? (y/n)y
SUCCESS: Added ipv4, ACCEPT rule for cnom, port 8080 on interface eth0
FAILED: ipv6, cnom is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for elasticsearch, port 8080 on interface e →
th0
FAILED: ipv6, elasticsearch is not resolvable. Provide valid service name/IP
SUCCESS: Added ipv4, ACCEPT rule for httpd, port 8080 on interface eth0
FAILED: ipv6, httpd is not resolvable. Provide valid service name/IP
Access allowed to services/IPs successfully
```

12.7.12.2 Removing Security for VNF-LCM with ENM on Cloud

This procedure is used to remove access towards VNF-LCM for the list of services provided. The details of services to be allowed are provided via file. The file contains a list of comma-separated service names, hostnames, or IPs.

Note: **--file <FILE>**

- Non-HA Scenario: If file is not provided, CLI will flush all the iptables rules
- Only text files(.txt) are allowed. Entries in a file must be comma-separated IPs or service names.
- To remove access of all services, run `removeaccess` CLI without file argument.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, then switch to root user.
2. To check for the existing services, use the list interface as mentioned in List Accessible iptable Rules towards VNF-LCM on page 230
3. Remove access towards VNF-LCM from the services, hostnames, or IPs specified in the input file:

```
[root@vnflaf-services ~]# vnflcm security removeaccess [--interface <eth0/et →
h1> --port <PORT> --file <FILE>]
```

Examples:

Note: Failure messages as in the following examples are expected when IPv6 is not supported.

Remove Access for Services:

```
[root@vnflaf-services ~]# vnflcm security removeaccess --interface eth1 --po →
rt 8080 --file /vnflcm-ext/enm/enm_iptables_white_list.txt
This CLI will remove traffic on port 8080 for IPs/services provided in file. →
 Are you sure to run the CLI? (y/n)y
SUCCESS: ipv4: Removed the ACCEPT rule for cnom, port 8080 on interface eth0
FAILED: ipv6: No rule match found in ip6tables to delete cnom, port 8080 on  →
interface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for elasticsearch, port 8080 on inter →
face eth0
FAILED: ipv6: No rule match found in ip6tables to delete elasticsearch, port →
 8080 on interface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for httpd, port 8080 on interface eth →
0
FAILED: ipv6: No rule match found in ip6tables to delete httpd, port 8080 on →
 interface eth0
Access removed for services/IPs successfully
```

Remove Access for Services : Port is not Provided and File is Provided

```
[root@vnflaf-services ~]# vnflcm security removeaccess --interface eth0 --fi →
le /vnflcm-ext/enm/enm_iptables_white_list.txt
This CLI will remove traffic on port 80 and 8080 for IPs/services provided i →
n file. Are you sure to run the CLI? (y/n)y
SUCCESS: ipv4: Removed the ACCEPT rule for cnom, port 80 on interface eth0
FAILED: ipv6: No rule match found in ip6tables to delete cnom, port 80 on in →
terface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for cnom, port 8080 on interface eth0
FAILED: ipv6: No rule match found in ip6tables to delete cnom, port 8080 on  →
interface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for elasticsearch, port 80 on interfa →
ce eth0
FAILED: ipv6: No rule match found in ip6tables to delete elasticsearch, port →
 80 on interface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for elasticsearch, port 8080 on inter →
face eth0
FAILED: ipv6: No rule match found in ip6tables to delete elasticsearch, port →
 8080 on interface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for httpd, port 80 on interface eth0
FAILED: ipv6: No rule match found in ip6tables to delete httpd, port 80 on i →
nterface eth0
SUCCESS: ipv4: Removed the ACCEPT rule for httpd, port 8080 on interface eth →
0
FAILED: ipv6: No rule match found in ip6tables to delete httpd, port 8080 on →
 interface eth0
Access removed for services/IPs successfully
```

Remove Access for Services : Port and File are not Provided / Port is Provided and File is not Provided:

```
[root@vnflaf-services ~]# vnflcm security removeaccess
File path not provided. CLI will reset all the iptables rules to default sta →
te. Are you sure to run the CLI? (y/n)y
Access removed for services/IPs successfully
```

12.7.12.3 List Accessible iptable Rules towards VNF-LCM

This procedure is used to list all of the security rules present in the `iptables`.

Note: List will display all the iptables rules added on VNF-LCM services via CLI or any other means.

Steps

1. Log on to `vnflaf-services` VM as cloud-user, then switch to root user.
2. List all of the security rules:

```
[root@vnflaf-services ~]# vnflcm security list
```

Example:

```
[root@vnflaf-services ~]# vnflcm security list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0            state RELATED, →
ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0            tcp dpt:12987
ACCEPT     tcp  --  161.160.141.60       0.0.0.0/0            tcp dpt:80
ACCEPT     tcp  --  161.160.141.60       0.0.0.0/0            tcp dpt:8080
ACCEPT     tcp  --  131.160.159.23       0.0.0.0/0            tcp dpt:80
ACCEPT     tcp  --  131.160.159.23       0.0.0.0/0            tcp dpt:8080
ACCEPT     tcp  --  141.159.162.36       0.0.0.0/0            tcp dpt:80
ACCEPT     tcp  --  141.159.162.36       0.0.0.0/0            tcp dpt:8080

DROP       all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

12.7.12.4 Applying Security for VNF-LCM with ENM on Cloud for Port 9002

> This section gives instructions to allow access between VNF-LCM VMs and ENM for port 9002 on external network.

Note: This section is relevant only if `ossType` in VNF-LCM SED is vENM.

Prerequisites

— VNF-LCM and ENM are successfully deployed.

— The required access details and keypair is available to log in.

— VNF-LCM is unable to receive trigger alarms from ENM on Cloud, or any workflow auto-trigger use case is failing.

Steps

1. Log in to VNF-LCM Services VM as cloud-user and switch to root user:

```
[cloud-user@vnflaf-services ~]# sudo su
```

   a. Create `vnflcm_external_iptables_white_list.txt` file with value for `<visinamingnb_external_ip_list>` and `<nbalarmirp_external_ip_list>` as defined on ENM SED:

```
[root@vnflaf-services ~]# echo <visinamingnb_external_ip_list>,<nba →
larmirp_external_ip_list> >> /vnflcm-ext/enm/vnflcm_external_iptabl →
es_white_list.txt
```

   Where `<visinamingnb_external_ip_list>` is external IP address of `visinamingnb` of ENM on cloud. This can be obtained from ENM on Cloud SED.

   `<nbalarmirp_external_ip_list>` is external IP address of `nbalarmirp` of ENM on cloud. This is obtained from ENM on Cloud SED.

   Example

```
[root@vnflaf-services ~]# echo 10.10.0.89,10.10.0.90 >> /vnflcm-ext →
/enm/vnflcm_external_iptables_white_list.txt
```

   b. Run the following command to add the ip to white list. This command allows access between VNF-LCM VMs and ENM for port `9002`:

```
[root@vnflaf-services ~]# vnflcm security allowaccess --interface e →
th1 --port 9002 --file /vnflcm-ext/enm/vnflcm_external_iptables_whi →
te_list.txt
```

2. Update `/etc/host` entries with `visinamingnb` ip address:

```
[root@vnflaf-services ~]# echo "<visinamingnb_external_ip_list>  visinamingn →
b   #added by vnflcm tool" >> /etc/hosts
```

   Where `<visinamingnb_external_ip_list>` is external IP address of `visinamingnb` of ENM on cloud. This is obtained from ENM on Cloud SED.

Example

```
[root@vnflaf-services ~]# echo "10.10.0.89  visinamingnb   #added by vnflcm
tool" >> /etc/hosts
```

3. Update `multivim_host` file present under `/vnflcm-ext/current/workflows` to backup the `visinamingnb` ip address:

```
[root@vnflaf-services ~]# echo "<visinamingnb_external_ip_list>  visinamingn
b   #added by vnflcm tool" >> /vnflcm-ext/current/workflows/multivim_host
```

Where `<visinamingnb_external_ip_list>` is external IP address of `visinamingnb` of ENM on cloud. This is obtained from ENM on Cloud SED.

Example

```
[root@vnflaf-services ~]# echo " 10.10.0.89  visinamingnb   #added by vnflcm
 tool" >> /vnflcm-ext/current/workflows/multivim_host
```

12.7.13 Add Private Network to VNF-LCM

VNFs on a private network appliance can be added to the VNF-LCM and ENM topology using the Add Private Network operation. The life cycle operations of the private network VNFs can be managed using VNF-LCM.

The Private Network (PN) is a pre-installed software consisting of 4G or 5G network (Core and RAN). The PN addresses the need of an organization (utilities, manufactures, transport companies) who want to have their on-premises 4G or 5G network (Core and RAN).

The network could be for data only or data and voice.

The on-premise private network is integrated with the local network of the enterprise and managed centrally by the Mobile operator (MNOs). A private network must integrate with the MNOs network management and orchestration processes and systems so that the MNOs can manage the lifecycle of the private network.

The present private network solution is offered with the following option:

— Enterprise Core (CEE/ OpenStack based)

— Multi-Purpose Server (VMware based)

— IMS/ VoLTE (VMware based)

Note:

— Add Private Network utility supports add or discovery of VNFs on Enterprise Core (CEE/ OpenStack based) appliance only as the VMware NFVI used in the Multi-Purpose Server and IMS/ VoLTE appliances does not include vCloudDirector.

— Multiple private network appliances can be added to VNF-LCM. However, private network appliance must be added one at a time.

Add Private Network utility support interactive and non-interactive modes. The following input files must be added to VNF-LCM for adding a private network.

Table 62

File Name	Description
Factory file	Factory file contains information of Virtual Infrastructure Manager (VIM) and VNFs that are part of a particular private network appliance. Factory file is delivered with a private network appliance. For more information, see Factory Input File Template on page 239.
Site file	Site File contains information about the ENM configuration. For more information, see Site Input File Template on page 241.
VNF file	VNF file contains the **`Managed Element ID`**, **`Node IP Address`**, **`subNetworks`**, and **`Mgmt/Supervision`** settings for each VNF in the appliance along with**`ApplianceName`** and **`Private Network Group Name`**. VNF file is mandatory if Add Private Network operation is performed in a non-interactive mode and optional in an interactive mode. For more information, see VNF Input File Template on page 243.
Configuration file	Configuration file contains the default configuration for private network VNF parameters like `transportProtocol` and `netConfPort` for each VNF type. The Configuration file is present in the path `/ericsson/vnflcm/data/pn_add_config.json`. For more information, see Configuration File on page 244.

Syntax

```
vnflcm privatenetwork add [-h] -f -s [-v] [-i]
```

Table 63 Command Arguments

Arguments	Description
`-h, --help`	Displays help information.
`-f, --factory`	JSON factory file delivered with Private Network Appliance.
`-s, --site`	JSON site file containing ENM configuration information.
`-v, --vnf`	JSON file containing VNF configuration in ENM such as `Managed Element ID`, `Node IP Address`, `subNetworks`.
`-i, --interactive`	Allows user to provide VNF file inputs interactively.

Non-interactive Mode

In a non-interactive mode:

— All the input files— Factory file, Site File, and VNF file are mandatory.

— All the input files are validated and if there are any validation errors, a consolidated error message is prompted with the details.

— Register the VIM in VNF-LCM if VIM is not added in VNF-LCM. During a VIM registration, JBoss server restarts.

— Add VNFs on the private network appliance to a VNF-LCM and ENM topology.

Interactive Mode

In an interactive mode:

— VNF file is optional.

— Factory file and Site file are validated and if there are any validation errors, a consolidated error message prompt with the details.

— If the VNF file is provided in the interactive mode, Add Private Network utility read the values provided in the VNF file, validate the values, and ask the user to provide values for the missing parameters containing invalid values in the VNF file.

— If VNF file is not provided, script collects the inputs for all the parameters present in VNF file.

— Register the VIM in VNF-LCM if VIM is not added in VNF-LCM. During a VIM registration, JBoss server restarts.

— Add VNFs on the private network appliance to a VNF-LCM and ENM topology.

Note: If the VIM mentioned in the Factory file is not already added in VNF-LCM, Add Private Network utility registers the VIM in VNF-LCM. During the VIM registration, JBOSS server restarts, and VNF-LCM service are unavailable during that time for a short duration.

Prerequisites

The following prerequisites are required to run the Add Private Network utility:

— Private Network Appliances are installed and connected to a network.

— ENM registered in EVNFM, a one-time activity.

— VNFM registered in ENM, a one-time activity.

— Private Network Groups are configured in ENM.

— The VNF packages and workflows are onboarded to VNF-LCM (for each release).

— Populate the Site Template (ENM configuration), a one time activity. For more information, see Site Input File Template on page 241.

12.7.13.1 Add Private Network to VNF-LCM - Non-Interactive Mode

> This section describe steps to Add Private Network to VNF-LCM in a non-interactive mode.

Prerequisites

See prerequisites for Add Private Network to VNF-LCM on page 232.

Steps

1. Retrieve the Factory file from the appliance and copy the file to the VNF-LCM service.
2. Copy the Site file to the VNF-LCM service.
3. Take a backup of original file `template_pn_vnf_input.json` and name the new file to `vnfInput.json`.

```
cp /ericsson/vnflcm/data/template_pn_vnf_input.json /tmp/vnfInput.json
```

4. Populate the `vnfInput.json` file for the private network appliance and copy it to VNF-LCM service. VNF file must contain the VNF details for each VNF present in the Factory file.
5. Verify the Configuration file present in the `/ericsson/vnflcm/data/pn_add_config.json` path, ensure that the Configuration file contains required configuration for each VNF type present in the Factory file.
6. Run add private network command.

```
[root@vnflaf-services-0 tmp]# vnflcm privatenetwork add --factory <factory_f
ile_path> --site <site_file_path> --vnf <vnf_input_file_path>
```

Example

```
[root@vnflafservices-0 tmp]# vnflcm privatenetwork add -f factoryMulti.json
-s factory_pn_site.json -v vnf3_nonInteractive.json
Validation passed. Starting operation...
VIM configuration not present in system, hence adding it
vim file saved in /ericsson/vnflcm/data/pnf_discovery/vim_file_Box1_20201005
_152546.json
Configuring certificates...
Default VIM parameters set to cloud configuration parameters
Restarting JBOSS...
```

```
Restarting jboss (via systemctl):                          [  OK  ]
+------+-------------------------+
| VIM  |          Result         |
+------+-------------------------+
| vim1 | VIM addition successful |
+------+-------------------------+
discovery file saved in /ericsson/vnflcm/data/pnf_discovery/discovery_file_B →
ox1_20201005_152737.json
Adding new vnf: vEPGtest
Starting OSS topology update for vnf vEPGtest
Oss topology updated with the vnf vEPGtest
VNF FDN value updation completed for vEPGtest
Adding new vnf: vEPGtest1
Starting OSS topology update for vnf vEPGtest1
Oss topology updated with the vnf vEPGtest1
VNF FDN value updation completed for vEPGtest1
Adding new vnf: vEPGtest2
Starting OSS topology update for vnf vEPGtest2
Oss topology updated with the vnf vEPGtest2
VNF FDN value updation completed for vEPGtest2
+--------------------------------------+---------------------------------- →
--+-----------------+--------------------------------------+
|               VnfId                  |               Vapp_Id               →
  | Operation_Status |               Message                |
+--------------------------------------+---------------------------------- →
--+-----------------+--------------------------------------+
| 34469cfe-06f1-11eb-a38f-fa163ef231d7 | be411bd5-9078-42c3-9bb3-645d29abedd →
5 |    Successful    | Update to VNF-LCM and OSS successful |
| 3f2de760-06f1-11eb-a38f-fa163ef231d7 | 91054eee-69ee-4d07-8916-2a9d991289c →
7 |    Successful    | Update to VNF-LCM and OSS successful |
| 4a19ecb1-06f1-11eb-a38f-fa163ef231d7 | 24454afc-c601-476c-a9e5-0b0409e010a →
0 |    Successful    | Update to VNF-LCM and OSS successful |
+--------------------------------------+---------------------------------- →
--+-----------------+--------------------------------------+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_152834.json
```

Results

Private network is added to VNF-LCM in a non-interactive mode.

12.7.13.2 Add Private Network to VNF-LCM - Interactive Mode

> In interactive mode, a private network can be added to VNF-LCM either with or without a VNF input file.

12.7.13.2.1 Add a Private Network without a VNF Input File: Interactive Mode

> If a VNF input file is not provided, the Add Private Network operation collects the inputs for all the parameters in the VNF file.

Prerequisites

See prerequisites for Add Private Network to VNF-LCM on page 232.

Steps

1. Retrieve the `Factory file` from the appliance and copy the file to VNF-LCM service.
2. Copy the `Site file` to the VNF-LCM service.

3. Verify the configuration file present in the `/ericsson/vnflcm/data/pn_add_config.json`. Ensure that the Configuration file contains required configuration for each VNF type present in the Factory file.

4. Run the Add Private Network utility command.

```
[root@vnflaf-services-0 tmp]# vnflcm privatenetwork add --factory <factory_f →
ile_path> --site <site_file_path> --interactive
```

Example

```
[root@vnflafservices-0 tmp]# vnflcm privatenetwork add --factory factoryMult →
i.json --site pn_site.json --interactive
Validation passed. Starting operation...
Collecting details to populate vnf input file. Enter CTRL+D to exit from the →
 operation.
Enter value for mandatory parameter applianceName: Box3
Enter value for mandatory parameter privateNetworkGroup: PNgroupepg
Enter value for optional parameter subNetworks: mySubNetwork
Enter value for optional parameter managementSupervisionSetting (false/true) →
. Default: false: false
Collecting details for vnf: vEPG_Test
Enter value for mandatory parameter managedElementId: vEPGtest
Enter value for optional parameter nodeIpAddress: 12.11.14.4
Finished collecting details for vnf: vEPG_Test
Collecting details for vnf: vEPG_Test1
Enter value for mandatory parameter managedElementId: vEOGtest1
Enter value for optional parameter nodeIpAddress: 12.11.12.6
Finished collecting details for vnf: vEPG_Test1
Collecting details for vnf: vEPG_Test2
Enter value for mandatory parameter managedElementId: vEPGtest2
Enter value for optional parameter nodeIpAddress: 12.11.13.2
Finished collecting details for vnf: vEPG_Test2
Finished collecting details for vnf input file.
VIM configuration not present in system, hence adding it
vim file saved in /ericsson/vnflcm/data/pnf_discovery/vim_file_Box3_20201005 →
_161653.json
Failed to backup host entry in /vnflcm-ext/current/workflows/multivim_host f →
ile.This is required for High Availiablity and Upgrade.Take backup of host e →
ntry manually
Configuring certificates...
Default VIM parameters set to cloud configuration parameters
Restarting JBOSS...
Restarting jboss (via systemctl):                          [  OK  ]
+------+-------------------------+
| VIM  |          Result         |
+------+-------------------------+
| vim1 | VIM addition successful |
+------+-------------------------+
discovery file saved in /ericsson/vnflcm/data/pnf_discovery/discovery_file_B →
ox3_20201005_161843.json
Adding new vnf: vEPGtest
Starting OSS topology update for vnf vEPGtest
Oss topology updated with the vnf vEPGtest
VNF FDN value updation completed for vEPGtest
Adding new vnf: vEOGtest1
Starting OSS topology update for vnf vEOGtest1
Oss topology updated with the vnf vEOGtest1
VNF FDN value updation completed for vEOGtest1
Adding new vnf: vEPGtest2
Starting OSS topology update for vnf vEPGtest2
Oss topology updated with the vnf vEPGtest2
VNF FDN value updation completed for vEPGtest2
+--------------------------------------+--------------------------------------- →
--+------------------+----------------------------------------+
|                VnfId                 |               Vapp_Id                  →
  | Operation_Status |               Message                  |
+--------------------------------------+--------------------------------------- →
--+------------------+----------------------------------------+
| 586593d0-06f8-11eb-88e6-fa163ef231d7 | be411bd5-9078-42c3-9bb3-645d29abedd →
5 |    Successful    | Update to VNF-LCM and OSS successful |
| 64ddb762-06f8-11eb-88e6-fa163ef231d7 | 91054eee-69ee-4d07-8916-2a9d991289c →
7 |    Successful    | Update to VNF-LCM and OSS successful |
| 6fdf8ea3-06f8-11eb-88e6-fa163ef231d7 | 24454afc-c601-476c-a9e5-0b0409e010a →
```

```
0 |    Successful    | Update to VNF-LCM and OSS successful |
+-------------------------------------+------------------------------------ →
--+-----------------+--------------------------------------+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_161945.json
```

Results

Private network is added to VNF-LCM in an interactive mode.

12.7.13.2.2 Add a Private Network with a VNF Input File: Interactive Mode

> If a VNF file is provided, the Add Private Network operation reads and validates the values in the VNF file. If any parameters are missing values or have invalid values, the operation requests that the correct values are added.

Prerequisites

See prerequisites for Add Private Network to VNF-LCM on page 232.

Steps

1. Retrieve the `Factory file` from the appliance and copy the file to VNF-LCM service.
2. Copy the Site file to VNF-LCM service.
3. Take a backup of original file `template_vnf.json` and name the new file to `vnfInput.json`.

   ```
   cp /ericsson/vnflcm/data/template_pn_vnf_input.json /tmp/vnfInput.json
   ```

4. Populate the `vnfInput.json` file for the private network appliance and copy it to VNF-LCM service. VNF file must contain the VNF details for each VNF present in the `Factory file`.
5. Verify the Configuration file present in the `/ericsson/vnflcm/data/pn_add_config.json` path. Ensure that the Configuration file contains the configuration for each VNF type present in factory file.
6. Run the Add Private Network utility command.

   ```
   [root@vnflaf-services-0 tmp]# vnflcm privatenetwork add --factory <factory_f →
   ile_path> --site <site_file_path> --vnf <vnf_input_file_path> --interactive
   ```

   Example

   ```
   Please noote in the following example, VNF file is provided, provided VNF  f →
   ile is validated and user was asked to provide values only for the missing p →
   arameters / parameters containing invalid values in the VNF file.

   [root@vnflafservices-0 tmp]# vnflcm privatenetwork add --factory factory_fil →
   ```

```
e.json --site pn_site.json --vnf vnfInput.json --interactive
Validation passed. Starting operation...
Collecting details to populate vnf input file. Enter CTRL+D to exit from the
 operation.
Enter value for mandatory parameter privateNetworkGroup: PNgroupepg
Enter value for optional parameter managementSupervisionSetting (false/true)
. Default: false: false
Collecting details for vnf: vEPG_Test2
Enter value for mandatory parameter managedElementId: Vepg_Test12
Enter value for optional parameter nodeIpAddress: 11.22.11.5
Finished collecting details for vnf: vEPG_Test2
Finished collecting details for vnf input file.
VIM configuration present in system. Skipping vim addition
discovery file saved in /ericsson/vnflcm/data/pnf_discovery/discovery_file_B
ox2_20201005_150031.json
Adding new vnf: vEPGtest2
Starting OSS topology update for vnf vEPGtest2
Oss topology updated with the vnf vEPGtest2
VNF FDN value updation completed for vEPGtest2
Adding new vnf: vEPGtest1
Starting OSS topology update for vnf vEPGtest1
Oss topology updated with the vnf vEPGtest1
VNF FDN value updation completed for vEPGtest1
Adding new vnf: Vepg_Test12
Starting OSS topology update for vnf Vepg_Test12
Oss topology updated with the vnf Vepg_Test12
VNF FDN value updation completed for Vepg_Test12
+--------------------------------------+--------------------------------------
--+-----------------+--------------------------------------+
|                VnfId                 |               Vapp_Id                
  | Operation_Status |              Message                 |
+--------------------------------------+--------------------------------------
--+-----------------+--------------------------------------+
| 6a085a01-06ed-11eb-82b9-fa163ef231d7 | be411bd5-9078-42c3-9bb3-645d29abedd
5 |    Successful    | Update to VNF-LCM and OSS successful |
| 757a7f43-06ed-11eb-82b9-fa163ef231d7 | 91054eee-69ee-4d07-8916-2a9d991289c
7 |    Successful    | Update to VNF-LCM and OSS successful |
| 80992d54-06ed-11eb-82b9-fa163ef231d7 | 24454afc-c601-476c-a9e5-0b0409e010a
0 |    Successful    | Update to VNF-LCM and OSS successful |
+--------------------------------------+--------------------------------------
--+-----------------+--------------------------------------+
Output saved in file /var/log/vnf_discovery/add_vnf_20201005_150129.json
```

Results

Private network is added to VNF-LCM in an interactive mode.

12.7.13.3 Json File Templates

> This section describes the JSON input file templates used for adding private network to VNF-LCM.

12.7.13.3.1 Factory Input File Template

This topic describes the Factory Input file template `template_pn_factory_input.json`, which is used by private network appliance to provide the details of VIM and VNFs that are part of a particular private network appliance. The file is delivered with private network appliance. The template file is stored at `/ericsson/vnflcm/data` folder.

```
{
    "vim": {
        "name": "",
        "type": "CEE",
        "hostIpAddress": "",
        "hostName": "",
```

```
        "authUrl": "",
        "domain": {
            "userDomain": "",
            "name": "",
            "id": "",
            "project": {
                "name": "",
                "id": "",
                "username": "",
                "password": ""
            }
        }
    },
    "vnfs": [{
        "vappName": "",
        "vappId": "",
        "vnfdId": "",
        "vnfVersion": "",
        "vnfType": "",
        "networkElementVersion": "",
        "nodeIpAddress": "",
        "nodeUsername": "",
        "nodePassword": ""
    }]
}
```

Table 64 Factory Input File Properties

Parameter Name			Data Type	Required (Yes/No)	Comment
vim					VIM details of the Private Network Appliance. Private Network Appliance has a single VIM.
		name	String	Yes	Name of the VIM where the Private Network VNFs are deployed.
		type	String	Yes	Type of VIM. Use one of these values— CEE, OpenStack, or VCD.
		hostIpAddress	String	Yes	IP address of the VIM.
		hostName	String	Yes	Hostname of the VIM. The entry is added in the Host file of VNF-LCM during the addition of VIM.
		authUrl	String	Yes	REST URL corresponding to the cloud authentication service.
	domain				Domain information for VIM. Private Network Appliance has a single VIM with a single domain.
		userDomain	String	No	Mandatory for OpenStack/CEE with v3 Keystone/Identity service. User Domain configured at cloud platform. Applicable for OpenStack/CEE with v3 Keystone/Identity service.
		name	String	No	Mandatory for OpenStack/CEE with v3 Keystone/Identity service. This parameter is provided with

Parameter Name				Data Type	Required (Yes/No)	Comment
						domain name configured in a cloud platform.
			id	String	No	Mandatory for OpenStack/ CEE with v3 Keystone/ Identity service. This parameter is provided with domain Id configured in a cloud platform.
		project				Project information for VIM.
			name	String	Yes	Project name configured in a cloud platform.
			id	String	Yes	Project Id configured in a cloud platform.
			username	String	Yes	User configured in a cloud platform for a project.
			password	String	Yes	Password of the user.
vnfs						Information of the VNFs that are present in the Private Network Appliance.
			vappName	String	Yes	The name of the vapp or stack on the VIM.
			vappId	String	Yes	The stack Id (in CEE and Openstack) or vappid (in VCD) of the VNF on the VIM.
			vnfdId	String	Yes	VNF descriptor Id. Identifier to identify the VNF package of the VNF.
			vnfVersion	String	No	Version of the VNF and an optional attribute. If not provided, it is auto-populated with the VNF version from the VNF package.
			vnfType	String	Yes	The type of the VNF. For example, vMME and vEPG.
			networkElementVersion	String	Yes	The Version of the Network Element. In ENM, this parameter refers to `ossModelIdentity`. It is mandatory for YANG-based nodes— vEPG (version higher than 2.0) CCSM, CCDM, CCRC, CCES, CCPC, PCC, PCG, SC.
			nodeIpAddress	String	Yes	The O&M IP Address of the Network Element.
			nodeUsername	String	Yes	Username for logging on to the VNF.
			nodePassword	String	Yes	Password for logging on to the VNF.

12.7.13.3.2 Site Input File Template

This section describe the Site Input file template `template_pn_site_input.json`, which contains the information about the ENM

configuration. User needs to create Site Input file by copying this template file before running the Adding Private Network operation, which is a one time activity. The template file is stored at `/ericsson/vnflcm/data` folder. Take a backup of original file `template_pn_site_input.json` while creating a `Site input` file.

```
{
    "snmpPort": "",
    "snmpVersion": "",
    "snmpSecurityLevel": "",
    "snmpSecurityName": "",
    "snmpAuthProtocol": "",
    "snmpPrivProtocol": "",
    "snmpAuthPassword": "",
    "snmpPrivacyPassword": "",
    "timeZone": "",
    "vnfmId": ""
}
```

Table 65 Site Input File Properties

Parameter Name	Data Type	Required (Yes/ No)	Comment
`snmpPort`	String	No	SNMP port is an optional parameter. If port is not provided, the default **`port 161`** is set.
`snmpVersion`	String	No	SNMP version is an optional parameter. In ENM for SNMP V3, it is mandatory to provide the value as **`SNMP_V3`**. For SNMP V2, it is mandatory to provide as **`SNMP_V2`**.
`snmpSecurityLevel`	String	No	Mandatory and applicable for SNMP V3. ENM supports three security levels— `AUTH_PRIV`, `AUTH_NO_PRIV`, `NO_AUTH_NO_PRIV`. Security level `NO_AUTH_PRIV` is not supported.
`snmpSecurityName`	String	No	Mandatory and applicable for SNMP V3.
`snmpAuthProtocol`	String	No	Authentication protocol used to authorize the identity of user. If not provided, it is to `MD5`. The parameter is applicable if SNMP V3 is used and the security level is either `AUTH_NO_PRIV` or `AUTH_PRIV`.
`snmpPrivProtocol`	String	No	Privacy Protocol used for encryption of SNMP v3 messages to ensure confidentiality of data. If not provided, it defaults to `AES128` for ENM. Applicable if SNMP V3 is used and the security level is `AUTH_PRIV`.
`snmpAuthPassword`	String	No	Mandatory when SNMP V3 is used and the security level is`AUTH_NO_PRIV` or `AUTH_PRIV`. Password set for authentication with the provided Authentication

Parameter Name	Data Type	Required (Yes/ No)	Comment
			Protocol for secure authorization.
snmpPrivacyPassword	String	No	Mandatory when SNMP V3 is used and the security level isAUTH_PRIV. Password set for authentication with the provided Authentication Protocol for secure authorization.
timeZone	String	No	If timeZone is not provided, then the timeZone of the system is set.
vnfmId	String	Yes	VirtualNetworkFunctionManager Id used for registering VNFM in ENM.

12.7.13.3.3 VNF Input File Template

This section describe the VNF input file template template_pn_vnf_input.json, which is used by user for providing Managed Element ID, Node IP Address, subNetworks and Mgmt/Supervision setting for each VNF in the appliance and the Appliance Name and Private Network Group name. VNF input file is mandatory if add private network operation performed in a non-interactive mode and optional for the interactive mode. The file is stored at /ericsson/vnflcm/data folder. Take a backup of original file template_pn_vnf_input.json while creating a VNF input file.

```
{
    "applianceName": "",
    "subNetworks": "",
    "managementSupervisionSetting": "",
    "privateNetworkGroup": "",
    "vnfs": [
        {
            "stackName": "",
            "managedElementId": "",
            "nodeIpAddress": ""
        }
    ]
}
```

Table 66 VNF Input File Properties

Parameter Name	Data Type	Required (Yes/ No)	Comment
applianceName	String	Yes	Name of the private network appliance.
subNetworks	String	No	SubNetwork is used for logical grouping of nodes within ENM. It can contain one or more child SubNetwork. For example, if only one subNetwork then the format must be: "subNetowrks": "CL" and if more than one child SubNetwork, then the format must be "subNetowrks": "CL,SubNetwork=COMS" That

Parameter Name		Data Type	Required (Yes/ No)	Comment
				is, SubNetwork must be separated by ",".
managementSupervisonSetting		String	No	The parameter contains **true** or **false**and the default value is **false**. This parameter is used to enable or disable the following supervision settings —pmFunction, cmNodeHeartbeatSupervision, and fmAlarmSupervision.
privateNetworkGroup		String	Yes	Collection name in ENM to group all Private Network VNFs.
Vnfs				VNF Information must contain all VNFs present in the factory file.
	stackName	String	Yes	vappName or stack name must match with vappName in the Factory input file.
	managedElementId	String	Yes	Recommended values for areManagedElementId either of the following: — vnfInstanceName — vappName — stackName
	nodeIpAddress	String	No	The O&M IP address of the Network Element, if not provided, nodeIpAddress from the Factory Input file is used.

12.7.13.4 Configuration File

This section describe the Configuration file. template pn_add_config.json. This Configuration file contains default values for - transportProtocol and netConfPort for VNF type. Before performing add private network operation, user must ensure that configuration file contains required configuration for each VNF type present in factory file. This configuration file is present in the /ericsson/vnflcm/data/pn_add_config.json path.

```
[
    {
        "vnfType": "MME",
        "transportProtocol": "SSH",
        "netConfPort": "22"
    },
    {
        "vnfType": "EPG",
        "transportProtocol": "SSH",
        "netConfPort": "830"
    },
    {
        "vnfType": "SAPC",
        "transportProtocol": "SSH",
        "netConfPort": "830"
    },
    {
        "vnfType": "HSS-FE",
        "transportProtocol": "SSH",
        "netConfPort": "830"
    },
```

```
    {
        "vnfType": "CUDB",
        "transportProtocol": "SSH",
        "netConfPort": "830"
    },
    {
        "vnfType": "EDA",
        "transportProtocol": "SSH",
        "netConfPort": "830"
    }
]
```

Table 67 Private Network Configuration File Properties

Parameter Name	Data Type	Required (Yes/ No)	Comment
`vnfType`	String	Yes	The parameter must match with `vnfType` in the Factory Input file and contains all the VNF types present in the Factory Input file.
`transportProtocol`	String	No	If `transportProtocol` is not provided, then the default value **SSH** is set. Possible values are **SSH** and **TLS**.
`netConfPort`	String	Yes	In ENM, for vEPG it is **830**, and for vMME it is **22**.

12.8 Workflow Bundle

A VNF Workflow Bundle is a versioned software package consisting of workflows required for managing life cycle of a VNF. This section describes the tasks related to Workflow Bundle management.

Note: In case of VNF-LCM HA deployment, use vip of services VM to access the system. Similarly internal vip needs to be used for DB VM. User is blocked to perform any activity on standby VM.

— RELATED INFORMATION —

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.8.1 List Workflow Bundles

List the workflow bundles that are installed and available for VNF-LCM.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is opened.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Run list command.

```
[root@vnflaf-services ~]# wfmgr bundle list [--name=<bundle_name>] [--versio →
n=<bundle_version>]
```

Example

List all bundles

```
[root@vnflaf-services tmp]# wfmgr bundle list
Name                | Version        | Package
--------------------+----------------+------------------------------------ →
-----------------------------------
vMMEWfBundle        | 0.0.1          | ERICvMMEWfBundle_CXP45128-0.0.1-20161 →
124151159.noarch.rpm
vMMEWfBundle        | 0.0.2          | ERICvMMEWfBundle_CXP45128-0.0.2-20161 →
124122138.noarch.rpm
vEPG-workflows      | 0.0.1          | ERICvEPG-workflows_CXP34ER.54-0.0.1-2 →
0160711101118.noarch.rpm
vEPG-workflows      | 1.0.1          | ERICvEPG-workflows_CXP34ER.54-1.0.1-2 →
0160718113232.noarch.rpm
vEPG-workflows      | 1.0.2          | ERICvEPG-workflows_CXP34ER.54-1.0.2-2 →
0160810151104.noarch.rpm
```

Example

List bundles by name

```
[root@vnflaf-services tmp]# wfmgr bundle list --name=vEPG-workflows
Name                | Version        | Package
--------------------+----------------+------------------------------------ →
-----------------------------------
vEPG-workflows      | 0.0.1          | ERICvEPG-workflows_CXP34ER.54-0.0.1-2 →
0160711101118.noarch.rpm
vEPG-workflows      | 1.0.1          | ERICvEPG-workflows_CXP34ER.54-1.0.1-2 →
0160718113232.noarch.rpm
vEPG-workflows      | 1.0.2          | ERICvEPG-workflows_CXP34ER.54-1.0.2-2 →
0160810151104.noarch.rpm
```

Example

List bundles by name and version

```
[root@vnflaf-services tmp]# wfmgr bundle list --name=vEPG-workflows --versio →
n=1.0.1
Name                | Version        | Package
--------------------+----------------+------------------------------------ →
-----------------------------------
vEPG-workflows      | 1.0.1          | ERICvEPG-workflows_CXP34ER.54-1.0.1-2 →
0160718113232.noarch.rpm
```

Results

Installed workflow bundles are listed.

12.8.2 Install Workflow Bundle

> Install the workflow bundle to be able to manage workflows using VNF-LCM UI.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is opened.

— Workflow RPM (Redhat Package Manager) bundle is downloaded to a directory inside vnflaf-services VM.

— Jboss/Server must be running on both master and standby VMs in case of HA setup.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Follow the steps on section List Workflow Bundles to make sure that bundle is not installed.
3. Install the workflow bundle package by running the install command.

```
[root@vnflaf-services ~]# wfmgr bundle install --package=<workflow_bundle_rp
m_file_path>
```

Example

```
[root@vnflaf-services ~]# wfmgr bundle install --package=/tmp/ERICvEPG_CXP45
128-2.1.26.rpm
Validating package...
Package validation done
VNFLCM services will not be availiable for few minutes
Installing package...
Preparing...                ########################################### [100
%]
RPM Preinstall
   1:ERICvEPG_CXP45128      ########################################### [100
%]
RPM Postinstall
Validating deployment for /tmp/ERICvEPG_CXP45128-2.1.26.rpm
vEPG-workflows-war-2.1.26.war
Successfully deployed workflows
+-----------------------------------+-------------+---------+--------------+
---------------------------------+
|             package               | pre_install | install | post_install |
            message             |
+-----------------------------------+-------------+---------+--------------+
---------------------------------+
| /tmp/ERICvEPG_CXP45128-2.1.26.rpm |   success   | success |   success    |
 package installation successful |
+-----------------------------------+-------------+---------+--------------+
---------------------------------+
```

Note: This example assumes that the rpm file is downloaded and placed under `/tmp` folder.

12.8.3 Uninstall Workflow Bundle

> Remove the workflows that are no longer required for VNF-LCM. Uninstalling workflows causes the JBOSS to restart, so the UI won't be accessible during this time.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is opened.

— Workflow bundle is already installed.

— Jboss/Server must be running on both master and standby VMs in case of HA setup.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to `root` user.
2. Follow the steps on section List Workflow Bundles to make sure that the relevant bundle is currently installed.
3. Uninstall the workflow bundle package.

```
[root@vnflaf-services ~]# wfmgr bundle uninstall --name=<bundle_name> --vers →
ion=<bundle_version>
```

4. The command will ask for user confirmation. Confirm to begin uninstall.

   Example

```
[root@vnflaf-services tmp]# wfmgr bundle uninstall --name=vEPG-workflows --v →
ersion=0.0.1
Warning: All current and historical information for workflow bundle will be  →
deleted
Are you sure you want to continue(yes/no)?
yes
Validating deployment...
Proceeding with bundle uninstall
Stopping jboss: *** JBossAS process (29036) received TERM signal ***
                                                           [  OK  ]
Starting jboss:                                 [  OK  ]
Package removed successfully
Uninstall successfully completed.
```

Results

Workflow bundle is successfully removed from the server.

12.8.4 List Bundle Descriptor

List the details of workflow bundles descriptor configured in VNF-LCM

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— Command console is open.

— Wfmgr bundle is already installed.

— Jboss/Server must be running.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Run list command

```
[root@vnflaf-services]# vnflcm bundle_descriptor list [--name<bundle_name>]
[--version=<bundle_version>]
```

Example

List all Workflow Bundle Descriptor.

```
[root@vnflaf-services]# vnflcm bundle_descriptor list
| name   | version
--------+--------
 vnflaf | 2.3.4
 vnflaf | 2.3.6
 vEPG   | 1.2.3
```

List all Workflow Bundle Descriptor by name

```
[root@vnflaf-services]# vnflcm bundle_descriptor list --name=vnflaf
 name   | version
--------+--------
 vnflaf | 2.3.4
 vnflaf | 2.3.6
```

List Workflow Bundle Descriptor by name and version

```
[root@vnflaf-services]# vnflcm bundle_descriptor list --name=vnflaf --versio
n=2.3.6
 name   | version
--------+--------
 vnflaf | 2.3.6
```

List Workflow Bundle Descriptor by ID

```
[root@vnflaf-services]# vnflcm bundle_descriptor list --name=vIMS_work --ver
sion=1.2.5 --details
================================================================================
```

```
==========

Workflow Bundle Descriptor Details:
ParameterName | Value
--------------+-------------
name          | vIMS_work
version       | 1.2.5
category      | VNFLCM


Workflow level Additional Parameters are:
ParameterName | Value
--------------+-------
key1          | value1
key2          | value2

Supported Use cases for Product :: vCSCF
------------------------------------------------------------------------------ →
----------

Use Case Property Details for workflow defination: Common_INSTANTIATE_def__t →
op

ParameterName         | Value
----------------------+---------------------------
name                  | INSTANTIATE
workflowDefinition    | Common_INSTANTIATE_def__top
minVnfSoftwareVersion | 1.16
maxVnfSoftwareVersion | *
emTriggerAllowed      | True
nfvoTriggerAllowed    | True
askForNfvoGrant       | True
notifyNfvo            | True


Use-case level Additional Parameters are:
ParameterName | Value
--------------+-------
key1          | value1
key2          | value2
------------------------------------------------------------------------------ →
----------

Use Case Property Details for workflow defination: Common_INSTANTIATE_def__t →
op

ParameterName         | Value
----------------------+---------------------------
name                  | INSTANTIATE
workflowDefinition    | Common_INSTANTIATE_def__top
minVnfSoftwareVersion | 1.14
maxVnfSoftwareVersion | 1.15


Use-case level Additional Parameters are:
ParameterName | Value
--------------+-------
key1          | value1
key2          | value2
------------------------------------------------------------------------------ →
----------

Use Case Property Details for workflow defination: Common_TERMINATE_def__top

ParameterName         | Value
----------------------+--------------------------
name                  | TERMINATE
workflowDefinition    | Common_TERMINATE_def__top
minVnfSoftwareVersion | 3.44
maxVnfSoftwareVersion | 2.55
nfvoTriggerAllowed    | True
askForNfvoGrant       | True


Use-case level Additional Parameters are:
ParameterName | Value
--------------+-------
key1          | value1
key2          | value2
```

```
Supported Use cases for Product :: vSBG
------------------------------------------------------------------------------ →
----------

Use Case Property Details for workflow defination: Common_INSTANTIATE_def__t →
op

ParameterName         | Value
----------------------+---------------------------
name                  | INSTANTIATE
workflowDefinition    | Common_INSTANTIATE_def__top
minVnfSoftwareVersion | *
maxVnfSoftwareVersion | *
emTriggerAllowed      | True
nfvoTriggerAllowed    | True
askForNfvoGrant       | True
notifyNfvo            | True


Use-case level Additional Parameters are:
ParameterName | Value
--------------+---------
key1          | value1
key2          | value2
test2         | testing2
test          | testing1
------------------------------------------------------------------------------ →
----------

Use Case Property Details for workflow defination: Common_SCALE_def__top

ParameterName         | Value
----------------------+----------------------
name                  | SCALE
workflowDefinition    | Common_SCALE_def__top
minVnfSoftwareVersion | *
maxVnfSoftwareVersion | *
emTriggerAllowed      | True
nfvoTriggerAllowed    | True
askForNfvoGrant       | True
notifyNfvo            | True


Use-case level Additional Parameters are:
ParameterName | Value
--------------+-------
key1          | value1
key2          | value2
------------------------------------------------------------------------------ →
----------

Use Case Property Details for workflow defination: Common_TERMINATE_def__top

ParameterName         | Value
----------------------+--------------------------
name                  | TERMINATE
workflowDefinition    | Common_TERMINATE_def__top
minVnfSoftwareVersion | *
maxVnfSoftwareVersion | *
emTriggerAllowed      | True
nfvoTriggerAllowed    | True
askForNfvoGrant       | True
notifyNfvo            | True


Use-case level Additional Parameters are:
ParameterName | Value
--------------+-------
key1          | value1
key2          | value2
============================================================================== →
==========
```

12.8.5 Update Bundle Descriptor

Updates values of workflow bundles descriptor configured in VNF-LCM. User cannot update bundle level details. Only additional parameters and use case definition properties can be updated.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is opened.

— Wfmgr bundle is already installed.

— Jboss/Server must be running.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Create Update bundle descriptor Json

   Create below Json to modify parameter of bundle descriptor.

   Note: User can modify the existing create bundle descriptor json present in `/opt/ericsson/ERIC<workflow_bundle_name>workflows/useCase-workflow-mapping/<bundle_name>/<version>/` directory.

```
{
    "category": "VNFLCM",
    "name": "BundleDescriptor",
    "version": "3.4.5",
    "supportedProducts": [{
        "name": "vEPG",
        "supportedPackageTypes": ["OVF","HOT","TOSCA"],
        "supportedUseCases": [{
                "name": "INSTANTIATE",
                "workflowDefinition": "Instantiate VNF-LAF v1",
                "minVnfSoftwareVersion": "*",
                "maxVnfSoftwareVersion": "*",
                "emTriggerAllowed": true,
                "nfvoTriggerAllowed": false,
                "askForNfvoGrant": false,
                "notifyNfvo": false,
                "additionalProperties": [{
                        "name": "key1",
                        "value": "value1"
                    },
                    {
                        "name": "key2",
                        "value": "value2"
                    }],
                "applicableUserRoles": ["VNLCMOperator", "VNLCMAdministrator →
", "vIMSOperator"]
            },
            {
                "name": "TERMINATE",
                "workflowDefinition": "Terminate VNF-LAF v1",
                "minVnfSoftwareVersion": "*",
                "maxVnfSoftwareVersion": "*",
                "emTriggerAllowed": true,
```

```
                "nfvoTriggerAllowed": false,
                "askForNfvoGrant": false,
                "notifyNfvo": false,
                "additionalProperties": [{
                        "name": "key1",
                        "value": "value1"
                    },
                    {
                        "name": "key2",
                        "value": "value2"
                    }],
                "applicableUserRoles": ["VNLCMOperator", "VNLCMAdministrator
", "vIMSOperator"]
            },
            {
                "name": "SCALE",
                "workflowDefinition": "Scale VNF-LAF v1",
                "minVnfSoftwareVersion": "*",
                "maxVnfSoftwareVersion": "*",
                "emTriggerAllowed": true,
                "nfvoTriggerAllowed": false,
                "askForNfvoGrant": false,
                "notifyNfvo": false,
                "additionalProperties": [{
                        "name": "key1",
                        "value": "value1"
                    },
                    {
                        "name": "key2",
                        "value": "value2"
                    }],
                "applicableUserRoles": ["VNLCMOperator", "VNLCMAdministrator
", "vIMSOperator"]
            }]
    }],
    "additionalProperties": [{
            "name": "key1",
            "value": "value1"
        },
        {
            "name": "key2",
            "value": "value2"
        }]
}
```

Description of parameters in JSON

Attribute	Required	Update Possible	Description
additionalProperties	N	Y	Key Value pairs. Additional properties required by use-case/workflow definition at the time of execution. This will be injected to workflow instance.
category	Y	N	Type ENUM [VNFLCM-For Lifecycle workflows of VNFs & R-VNFM, APPLCM-For application workflows e.g. vENM workflows, VNFMCLIENT-For workflows triggering external workflows e.g. RAN workflows].
name	Y	N	Workflow bundle name. Alphanumeric + some special

Attribute	Required	Update Possible	Description
			characters allowed as defined in SDK maven archetype.
supportedProducts	N	Y	List of all products supported by the workflow bundle
supportedProducts.name	Y	N	Supported product name. Alphanumeric string.
supportedProducts.supportedPackageTypes	Y	Y	List of supported packages. Possible values are OVF, HOT and TOSCA.
supportedProducts.supportedUseCases	Y	Y	List of Use case definitions. Use case name + workflowDefinition + supported SW versions together defines the uniqueness of the Use case structure in the list.
supportedProducts.supportedUseCases.maxVnfSoftwareVersion	Y	N	Minimum and maximum product software versions supported by the workflow definition. Following rules are applied by VNFLCM. Example scenario where 1.14, 1.15, 1.16 and 1.17 are vnf software versions available and to be supported by workflows "minVnfSoftwareVersion": "1.15", //oldest VNF release supported by the workflow is 1.15. "maxVnfSoftwareVersion": "1.17", //latest VNF release supported by the workflow is 1.17. Overall 1.15, 1.16 and 1.17 vnf software versions are supported by the workflow. "minVnfSoftwareVersion": "*", //wild card value here means any version lower than max version is supported "maxVnfSoftwareVersion": "1.16", //latest VNF release

Attribute	Required	Update Possible	Description
			supported by the workflow is 1.17. Overall 1.14, 1.15 and 1.16 vnf software versions are supported by the workflow. "minVnfSoftwareVersion": "1.16", //oldest VNF release supported by the workflow is 1.16. "maxVnfSoftwareVersion": "*", //wild card value here means any version greater than min version is supported. Overall 1.16 and 1.17 vnf software versions are supported by the workflow. "minVnfSoftwareVersion": "*", //wild card value here means any version lower than max version is supported. "maxVnfSoftwareVersion": "*", //wild card value here means any version greater than min version is supported. Overall all 1.14, 1.15, 1.16 and 1.17 vnf software versions are supported by the workflow.
`supportedProducts.supportedUseCases.minVnfSoftwareVersion`	Y	N	Minimum and maximum product software versions supported by the workflow definition. Following rules are applied by VNFLCM. Example scenario where 1.14, 1.15, 1.16 and 1.17 are vnf software versions available and to be supported by workflows "minVnfSoftwareVersion": "1.15", //oldest VNF release supported by the workflow is 1.15. "maxVnfSoftwareVersion": "1.17", //latest VNF release

Attribute	Required	Update Possible	Description
			supported by the workflow is 1.17. Overall 1.15, 1.16 and 1.17 vnf software versions are supported by the workflow. "minVnfSoftwareVersion": "*", //wild card value here means any version lower than max version is supported "maxVnfSoftwareVersion": "1.16", //latest VNF release supported by the workflow is 1.17. Overall 1.14, 1.15 and 1.16 vnf software versions are supported by the workflow. "minVnfSoftwareVersion": "1.16", //oldest VNF release supported by the workflow is 1.16. "maxVnfSoftwareVersion": "*", //wild card value here means any version greater than min version is supported. Overall 1.16 and 1.17 vnf software versions are supported by the workflow. "minVnfSoftwareVersion": "*", //wild card value here means any version lower than max version is supported. "maxVnfSoftwareVersion": "*", //wild card value here means any version greater than min version is supported. Overall all 1.14, 1.15, 1.16 and 1.17 vnf software versions are supported by the workflow.
`supportedProducts.supportedUseCases.name`	Y	N	Name of the use case.

Attribute	Required	Update Possible	Description
supportedProducts.supportedUseCases.nfvoTriggerAllowed	N	Y	Boolean value. If true then workflow supports start via NBI. Required only if category is VNFLCM. This flag must be set to "true" where workflow is started from the GUI in Full stack mode.
supportedProducts.supportedUseCases.notifyNfvo	N	Y	Boolean value. If false then workflow will not ask for grant from NFVO. Required only if category is VNFLCM.
supportedProducts.supportedUseCases.workflowDefinition	Y	N	Mapped workflow definition name. Aplha numeric strings. Spaces allowed.
version	Y	N	Workflow bundle version. Supported version format is x.y.z.

3. Update the workflow bundle descriptor by running the following command.

```
[root@vnflaf-services]# vnflcm bundle_descriptor update --file=<file_name>
```

Example

```
[root@vnflaf-services]# vnflcm bundle_descriptor update --file=/UpdateWorflo →
wBundleDescriptor.json
Bundle descriptor updated successfully
```

12.8.6 List Workflow ProcessId

List Workflow ProcessId along with corresponding Definition Name that are installed and available for VNF-LCM.

Prerequisites

— Root and cloud-user access to the vnflaf-services VM.

— A command console is opened.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Run list command:

```
[root@vnflaf-services ~]# wfmgr workflow processid-list [--name=<bundle_name →
>] [--version=<bundle_version>]
```

Example

List all processIds:

```
[root@vnflaf-services ~]# wfmgr workflow processid-list
+---------------+---------+---------------------------+-------------------- →
----+
| Name          | Version | ProcessId                 | DefinitionName      →
    |
+---------------+---------+---------------------------+-------------------- →
----+
| stackCreation | 2.4.23  | createStack_V1__top       | createStack         →
    |
| stackCreation | 2.4.23  | Terminate__Process_V1__top | terminateStack      →
    |
| vDSC          | 1.1.0   | dscScaleIn_V1__top        | vDSC Scale In Node  →
    |
| vDSC          | 1.1.0   | vdscHeal_V1__top          | vDSC Heal Node      →
    |
| vDSC          | 1.1.0   | vdscUpgrade_V1__top       | vDSC Upgrade Node   →
    |
| vDSC          | 1.1.0   | vdscTerminate_V1__top     | vDSC Terminate Node →
    |
| vDSC          | 1.1.0   | vdscInstantiate_V1__top   | vDSC Instantiate No →
de  |
| vDSC          | 1.1.0   | dscScaleOut_V1__top       | vDSC Scale Out Node →
    |
+---------------+---------+---------------------------+-------------------- →
----+
```

Example

List ProcessIds by name:

```
[root@vnflaf-services ~]# wfmgr workflow processid-list --name=stackCreation
+---------------+---------+---------------------------+----------------+
| Name          | Version | ProcessId                 | DefinitionName |
+---------------+---------+---------------------------+----------------+
| stackCreation | 2.4.23  | createStack_V1__top       | createStack    |
| stackCreation | 2.4.23  | Terminate__Process_V1__top | terminateStack |
+---------------+---------+---------------------------+----------------+
```

Example

List ProcessIds by name and version

```
[root@vnflaf-services ~]# wfmgr workflow processid-list --name=vnflaf --vers →
ion=2.4.23
+--------+---------+---------------------------+------------------------+
| Name   | Version | ProcessId                 | DefinitionName         |
+--------+---------+---------------------------+------------------------+
| vnflaf | 2.4.23  | ScaleVNF-LAF-v1__top      | Scale VNF-LAF v1       |
| vnflaf | 2.4.23  | TerminateVNF-LAF-v1__top  | Terminate VNF-LAF v1   |
| vnflaf | 2.4.23  | HealVNF-LAF-v1__top       | Heal VNF-LAF v1        |
| vnflaf | 2.4.23  | InstantiateVNF-LAF-v1__top | Instantiate VNF-LAF v1 |
+--------+---------+---------------------------+------------------------+
```

Result: ProcessIds of the installed workflows are listed.

12.8.7 Disable a Workflow

Disable a workflow for any further operations.

Prerequisites

- Root and cloud-user access to the vnflaf-services VM.
- A command console is opened.
- Workflow bundle is already installed.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.
2. Follow the steps in List Workflow ProcessId on page 257 to get the ProcessId of the workflow to be disabled.
3. Disable the workflow by running the disable command.

```
[root@vnflaf-services ~]# wfmgr workflow disable [--name=<bundle_name>] [--v
ersion=<bundle_version>] [--processId=<process_id>]
```

Example

```
[root@vnflaf-services ~]# wfmgr workflow disable --name=vDSC --version=1.1.0
 --processId=dscScaleIn_V1__top
Workflow disabled successfully
```

Result: Required workflow is disabled.

12.8.8 Enable a Workflow

Enable a workflow to perform any operation.

Prerequisites

- Root and cloud-user access to the vnflaf-services VM.
- A command console is opened.
- Workflow bundle is already installed.

Steps

1. Log on to vnflaf-services VM as cloud-user and switch to root user.

2. Follow the steps in List Workflow ProcessId on page 257 to get the ProcessId of the workflow to be enabled.
3. Enable the workflow by running the enable command.

```
[root@vnflaf-services ~]# wfmgr workflow enable [--name=<bundle_name>] [--ve
rsion=<bundle_version>] [--processId=<process_id>]
```

Example

```
[root@vnflaf-services ~]# wfmgr workflow enable --name=vDSC --version=1.1.0
--processId=dscScaleIn_V1__top
Workflow enabled successfully
```

Result: Required workflow is enabled.

12.8.9 Package Installation Troubleshooting

If package installation is failing due to timeout issue in war deployment, then this time limit can be increased. It will make package installation successful.

Prerequisites

- Root and cloud-user access to the vnflaf-services VM.
- A command console is opened.
- Wfmgr bundle is already installed.
- Package should not be faulty.
- Jboss/Server must be running.

Steps

1. Run the following command:

```
vim /opt/ericsson/ERICwfmgrruntimetools_CXP9032765/waf_tools/service/constan
ts.py
```

2. Change the value of : **max_deployed_attempt**.

   For example: **max_deployed_attempt=6**

   If max_deployed_attempt = 1 , then total time is 20 seconds.

   If max_deployed_attempt = 6 , then total time is 6 x 20 sec = 120 sec.

12.9 Configure Rules Mapping File

> VNF-LCM NBI feature requires the workflow-mapping rules to be defined and configured correctly in mappings.xml. The correct definition and configuration ensures that VNF-LCM can execute a particular workflow for a particular use case for a given VNF type and version.

Note: For a VNF-LCM HA deployment, use the VIP of the services VM to access the system. Similarly, use the internal VIP for DB VM. Perform this activity on both master and standby VMs to avoid inconsistencies.

The `mapping.xml` file contains the mappings rules between the following:

— A use case

For example, Instantiation, Termination.

— The workflow definition

The workflow to be executed for that particular use case.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VMs

— A command console is opened.

— Integration with Openstack as VIM.

— The mappings.xml is in the `/etc/opt/ericsson/ERICvnflcmservice_CXP9033884/` directory.

Steps

1. Log on to `vnflaf-services` VM as cloud-user and switch to root user.
2. Update the mapping file according to elements described in section Mapping File Elements.

```
[root@vnflaf-services ~]#vi /etc/opt/ericsson/ERICvnflcmservi →
ce_CXP9033884/mappings.xml
```

3. Verify the Status to check NBI rule loading:

   A status file corresponding to the descriptor is created in:`/var/log/nbiDescriptorStatus/`.

   — `mappings.xml.isDeploying`

— Rule loading is in process.

— `mappings.xml.deployed`

  Rule loaded successfully.

— `mappings.xml.failed`

  Failed to load rule. File contains the reason for the failure.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.9.1 Mapping File Elements

Table 68 Rules-Mapping File Elements

Element	Description
<wft:vnfProductName>	Contains a value which is part of the attribute `vnfdId` (value before __) in the `POST` request for `INSTANTIATION`. Example: `mme__16B`. Here MME is the `<wft:vnfProductName>`. The same value is used for `TERMINATION`.
<wft:operation>	The life cycle operation for VNF. Valid values: `INSTANTIATION` or `TERMINATION`.
<wft:vnfSoftwareVersion>	Contains a value which is part of the attribute `vnfdId` (value after __) in the `POST` request for `INSTANTIATION`. Example: `mme__16B . 16B` is the `<wft:vnfSoftwareVersion>`. The same value is used for `TERMINATION`.
<wft:workflowInfo>	`workflowDefinitionName`. The definition name of the workflow that defines the action to be taken when the condition is matched. For example, `MME_Instantiate` Its attributes are: — `workflowBundleName` The name of the workflow application like "vMME". — `workflowBundleVersion`

Element	Description
	The workflow version.

Example of the mappings.xml File

```
Defines the action to be taken when the condition is matched. It →
s<?xml version="1.0" encoding="UTF-8"?>
                <wft:workflowMappingRules name="" xmlns:wft="urn →
:com:ericsson:schema:xml:oss:wft" xmlns:xsi="http://www.w3.org/2 →
001/XMLSchema-instance" xsi:schemaLocation="urn:com:ericsson:sch →
ema:xml:oss:wft mapping-rules.xsd ">
                <wft:workflowMappingRule>
                <wft:condition>
                <wft:applicationType>LCM</wft:applicationType>
                <wft:vnfProductName>mme</wft:vnfProductName>
                <wft:operation>INSTANTIATION</wft:operation>
                <wft:vnfSoftwareVersion>16B</wft:vnfSoftwareVers →
ion>
                </wft:condition>
                <wft:workflowInfo workflowBundleName="Random-Pas →
sword" workflowDefinitionName="<workflow_def_name>" workflowBund →
leVersion="0.0.1-SNAPSHOT"/>
                </wft:workflowMappingRule>
                <wft:workflowMappingRule>
                <wft:condition>
                <wft:applicationType>LCM</wft:applicationType>
                <wft:vnfProductName>mme</wft:vnfProductName>
                <wft:operation>TERMINATION</wft:operation>
                <wft:vnfSoftwareVersion>16B</wft:vnfSoftwareVers →
ion>
                </wft:condition>
                <wft:workflowInfo workflowBundleName="Random-Pas →
sword" workflowDefinitionName="<workflow_def_name>" workflowBund →
leVersion="0.0.1-SNAPSHOT"/>
                </wft:workflowMappingRule>

                </wft:workflowMappingRules>
```

12.10 Identify Workflows for Auto-Start Rules

Identify the workflow `definitionid` attribute of the workflow to be configured in the Rule XML file.

Note: For a VNF-LCM HA deployment, use VIP of services VM to access the system. Similarly, use an internal VIP for DB VM. Do the activity on both master and standby VMs to avoid inconsistencies.

Refer to section Sample Rule XML File to learn more about this file.

Steps

1. Open the `vnflaf-service` VM console and get the details of the necessary workflow in the auto-start rules file:

```
curl http://localhost:8080/wfs/rest/definitions
```

Result: An output similar to this is displayed:

```
[{"definitionId":"vMME.--.0.0.1.--.instantiateNode_Process_V1__top","definit →
ionName":"instantiate_node_process_version-1","description":null}

{"definitionId":"queryVappCheck.--.1.0.0.--.queryVapp","definitionName":"Val →
idate
 Virtual Application","description":null}
       {"definitionId":"queryVappCheck.--.1.0.0.--.authenticateCloud","defin →
itionName":"authenticateCloud","description":null},
       {"definitionId":"queryVappCheck.--.1.0.0.--.testAuth_v1__top","defini →
tionName":"testAuth_v1","description":null},
       {"definitionId":"queryVappCheck.--.1.0.0.--.queryVDC","definitionName →
":"queryVDC","description":null},
       {"definitionId":"alarmautotest.--.0.0.1.--.alarmauto__top","definitio →
nName":"alarm auto","description":null},

{"definitionId":"package-test.--.0.0.1.--.packagetest_Process_V1__top","defi →
nitionName":"Validate vnf package test","description":null},
```

2. Analyze the output to identify the self-healing workflow entry. Take note of the `definitionName` attribute:

```
{"{"definitionId":"alarmautotest.--.0.0.1.-.alarmauto__top","definitionName" →
:"alarm auto","description":null},
```

Result: From this, you can get the value of the `WorkflowBundleName` as `alarmautotest` and `definitionName` as alarm auto.

— RELATED INFORMATION —

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.10.1 Configure Auto-Start Rules for Alarms

> Configure the triggers in Auto Start feature to initiate LCM operations, such as scaling and healing, when specific alarms occur.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— A command console is opened.

Steps

1. Copy the XML template file from `/vnflcm-ext/current/workflows/auto-start-rules/` to a temporary directory, and configure the sample rules to run the Auto-Start Workflow.

   Refer to section Sample Rule XML File to learn more about this file.

2. Update the `triggerCondition` tag to compare the Alarm or Event based on which the LCM workflow has to be started:

   Example

   Define the Alarm-based trigger.

   ```
   <wft:triggerCondition>
       <wft:attributeName>AlarmId</wft:attributeName>
       <wft:triggerConditionOperator>EQUALS</wft:triggerConditionOperator>
       <wft:attributeValue>5423</wft:attributeValue>
   </wft:triggerCondition>
   ```

   The 'AlarmId' retrieved from the Alarm is compared with the attributeValue that is '5423.'

   Example

   Define the Event-based trigger.

   ```
   <wft:triggerCondition>
        <wft:attributeName>specificProblem</wft:attributeName>
        <wft:triggerConditionOperator>CONTAINS</wft:triggerConditionOperator>
        <wft:attributeValue>Active</wft:attributeValue>
   </wft:triggerCondition>
   ```

   The 'specificProblem' retrieved from the Alarm or Event is compared with the attributeValue that is 'Active'.

3. Update the `triggerAction` tag.

   This tag defines the action that occurs when the condition is met. In this case, it determines is which LCM workflow must be started.

   Example

   ```
   <wft:triggerAction WorkflowDefinitionName="Auto Start Workflow" WorkflowBund →
   leName="AutoStart"><wft:param />
   </wft:triggerAction>
   ```

   In this example, `definitionname` of the workflow is "Auto Start Workflow" and the bundle name is "AutoStart".

4. Copy the XML file from the temporary directory back to `/vnflcm-ext/current/workflows/auto-start-rules/` directory.

12.10.1.1 Sample Rule XML File

> The sample rules are specified in an XML file, named like *<workflowBundleName>*`-autostart-rules.xml`.

Attributes marked with * are mandatory.

<wft:workflowTriggerRule>

This is the rule element, which can have multiple triggerConditions and a triggerAction. Its attributes are:

— `*id- user`: The defined rule id.

— `*name- user`: The defined rule name.

— `*type- rule`: The trigger type, either `FM_ALARM` or `FM_EVENT`.

Note: The trigger type `FM_ALARM` indicates rules that are applicable for alarm only. The trigger type `FM_EVENT` indicates rules that are applicable for event only.

<wft:triggerCondition>

This element defines the condition based on which the action is performed.

Each triggerCondition has the following child elements:

<wft:attributeName>

This element is mapped with the Alarms and Event attributes.

The valid or the supported values for this element are `AlarmId`, `AlarmState`, `problemText`, `problemDetail`, `probableCause`, `managedObject`, `specificProblem`, `additionalInformation`.

<wft:triggerConditionOperator>

Describes the condition matching operation on the attribute values specified. Valid values for the element are `EQUALS`, `NOTEQUALS`, and `CONTAINS`.

— `EQUALS`

Attribute value received matches the same as configured value in the XML file.

- NOTEQUALS

  Attribute value does not match the defined value in the XML file.

- CONTAINS

  Attribute value contains the values defined in the XML file.

<wft:attributeValue>

This element defines the exact values to be evaluated.

For example, if `alarmId` attribute has value `12345`:

- `attributeName` is `AlarmId`.
- `attributeValue` is `12345` .
- `triggerConditionOperator` can be any valid value that is listed.

<wft:triggerAction>

This element defines the action to be taken when the condition is matched. Its attributes are:

- `*WorkflowDefinitionName` .

  The definition name of the workflow like "EPG_Instantiate".

- `*WorkflowBundleName` .

  The name of the workflow application like "vEPG".

Rule XML File Example

```
<?xml version="1.0" encoding="UTF-8"?>
<wft:workflowTriggerRules name="" type=""
        version="" xmlns:wft="urn:com:ericsson:schema:xml:oss:wft" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:com:ericsson:schema:xml:oss:wft auto-start-wf-rules.xsd ">
        <wft:workflowTriggerRule name="ruleName" id="ruleId" type="FM_ALARM">
                <wft:triggerCondition>
                        <wft:attributeName>AlarmId</wft:attributeName>
                        <wft:triggerConditionOperator>EQUALS</wft:triggerConditionOperator>
                        <wft:attributeValue>5423</wft:attributeValue>
                </wft:triggerCondition>
                <wft:triggerAction WorkflowDefinitionName="Auto Start Workflow"
                        WorkflowBundleName="AutoStart">
                        <wft:param />
                </wft:triggerAction>
        </wft:workflowTriggerRule>
```

12.11 VNF-LCM Autorecovery

Note: It is recommended to use `vnflcm` tool to enable or disable auto-recovery only for hardware maintenance ,firmware upgrades, and other maintenance activities.

Where the auto-recovery workflow is executed, user need to reset the password for the respective recovered VM.

If DB standby goes down due to compute node failure, make sure postgres replication is complete after standby recovery workflow is `100%`.

If `enable_autorecovery` parameter value is set to "True"/"False" in VNFLCM SITE ENGINEERING DATA while deployment, the parameter value will be overwritten with the deployment even if we disable it at runtime at the times of recovery and upgrade

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.12 Configuring rsyslog on VNF-LCM Services

> VNFLCM can be configured to send logs to a remote Centralized logging Server. VNF-LCM uses rsyslog Linux daemon to send the logs to the remote server and expects the remote server to use rsyslog daemon to receive the logs.

Complete the following steps to configure VNFLCM to send the logs.

Prerequisites

— Access to vnflcm VMs.

— rsyslogd service is running on vnflcm VMs both services and db.

— Remote server is also configured with rsyslogd service.

Steps

1. Login to VNFLCM Services VM as cloud-user and switch user to root.
2. Replace the `/etc/rsyslog.conf` file with the following:

```
# rsyslog configuration file
# note that most of this config file uses old-style format,
```

```
# because it is well-known AND quite suitable for simple cases
# like we have with the default config. For more advanced
# things, RainerScript configuration is suggested.
# For more information see /usr/share/doc/rsyslog-*/rsyslog_conf.html
# If you experience problems, see http://www.rsyslog.com/doc/troubleshoot.ht →
ml

#### MODULES ####
module(load="imuxsock") # provides support for local system logging (e.g. vi →
a logger command)
module(load="imklog")   # provides kernel logging support (previously done b →
y rklogd)

#### GLOBAL DIRECTIVES ####
# Use default timestamp format
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileFormat
# Include all config files in /etc/rsyslog.d/
$IncludeConfig /etc/rsyslog.d/*.conf

#### RULES ####
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages
# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  /var/log/maillog

# Log cron stuff
cron.*                                                  /var/log/cron
# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*
# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler
# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log

# ### begin forwarding rule ###
# The statement between the begin ... end define a SINGLE forwarding
# rule. They belong together, do NOT split them. If you create multiple
# forwarding rules, duplicate the whole block!
# Remote Logging (we use TCP for reliable delivery)
#
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.
$WorkDirectory /var/lib/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g   # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList   # run asynchronously
$ActionResumeRetryCount -1    # infinite retries if host is down
# remote host is: name/ip:port, e.g. 192.168.0.1:514, port optional
#*.* @@remote-host:514


*.* @LogServerHostName/IP:514

# ### end of the forwarding rule ###
```

Note:
- Replace the LogServerHostName/IP with the Hostname or IP Address of the Centralized Log Server where VNFLCM has to send the logs.
- If IPv6 Address is used, the IP addressed has to be enclosed in [ ] brackets. e.g. [2001:0db8:85a3:0000:0000:8a2e:0370:7334]:514
- If hostname is used, then make sure that the host entry is made with the correct IP address mapping in /etc/hosts file.
- The default port is 514 but if the rsyslog daemon on the remote log server is configured on a different port, change the port value accordingly.
- If TCP is used instead of UDP then @@ has to be used before the hostname/IP instead of a single @ symbol.

3. Remove any existing files in `/etc/rsyslog.d/` directory.

```
[root@vnflaf-services ~]# rm -rf /etc/rsyslog.d/*
```

4. Create the following files in `/etc/rsyslog.d/` directory :

   a. Create file named `0_base_log.conf` with the following contents:

```
$MaxMessageSize 65536
#RSYSLOG_ForwardFormat template is used to enable millisecond-accur →
acy
$ActionForwardDefaultTemplate RSYSLOG_ForwardFormat
#Uncomment below 2 lines if TCP is used.
#module(load="imtcp")
#input(type="imtcp" port="514")
#Comment below 2 lines if TCP is used.
module(load="imudp")
input(type="imudp" port="514")
mail.info stop
```

   Note: Change the port number if the rsyslog daemon on the remote log server is running on a different port.

   If the remote log server is using TCP instead of UDP, then uncomment the tcp module related lines and comment the udp module related lines as described in the inline comments.

   b. Create file named `10_vnflcm_services_log.conf` with the following contents:

```
$ModLoad imfile # needs to be done just once


$InputFileName /ericsson/3pp/jboss/standalone/log/server.log
$InputFileTag vnflcm-jboss:
$InputFileStateFile vnflcm-jboss
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
```

```
$InputFilePollInterval 1


$InputFileName /var/log/vnflcm-admin-cli/logfile.log
$InputFileTag vnflcm-admin-cli:
$InputFileStateFile vnflcm-admin-cli
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /var/log/wfmgr-cli-log/logfile.log
$InputFileTag vnflcm-wfmgr-cli:
$InputFileStateFile vnflcm-wfmgr-cli
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /vnflcm-ext/current/logs/vnflcm_backup.log
$InputFileTag vnflcm-backup:
$InputFileStateFile vnflcm-backup
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /vnflcm-ext/current/logs/vnflcm_restore.log
$InputFileTag vnflcm-restore:
$InputFileStateFile vnflcm-restore
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /var/log/vnflcm/.svc_check.log
$InputFileTag svc-check:
$InputFileStateFile svc-check
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /var/log/vnflcm/configure.log
$InputFileTag svc-check:
$InputFileStateFile svc-check
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /var/log/vnflcm/.health_check.log
$InputFileTag svc-health_check:
$InputFileStateFile svc-health_check
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /var/log/vnflcm/.recovery.log
$InputFileTag svc-recovery:
$InputFileStateFile svc-recovery
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1

$InputFileName /var/log/vnflcm/configure_per_boot.log
$InputFileTag svc-per-boot:
$InputFileStateFile svc-per-boot
$InputFileSeverity error
$InputFileFacility user
$InputRunFileMonitor
$InputFilePollInterval 1
```

Note: Only error logs are sent by default to the remote log server. If you need info logs as well then change the "error" in the `$InputFileSeverity` to info.

5. Restart `rsyslog` service:

```
[root@vnflaf-services ~]# service rsyslog restart
```

Note: VNF-LCM DB does not have an external network interface. Therefore, rsyslog configuration can not happen.

RELATED INFORMATION

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.13 Package On-Boarding in VNF-LCM

The method of on-boarding VNF packages depends on whether a Network Functions Virtualization Orchestrator (NFVO) is deployed.

Packages are on-boarded on VNF-LCM under `/vnflcm-ext/current/vnf_package_repo/`, which is required for workflow execution.

Prerequisites

— Root and cloud-user access to the `vnflaf-services` VM.

— A command console is open.

If `NVFO` is not deployed, see Small-Stack on page 272.

If `NVFO` is deployed, see Full-Stack on page 272.

Small-Stack

In this scenario, the packages are manually on-boarded to VNF-LCM.

The package is placed under `/vnflcm-ext/current/vnf_package_repo/{VNFD_ID}`, where {VNFD_ID} is the `VNFD ID` for the VNF.

Full-Stack

In this scenario, the packages are on-boarded in one of the following ways:

— Manually as in Small-Stack on page 272.

— Automatically:

  In this case, the on-boarding is handled as part of the `Instantiate LCM` operation. When the NFVO calls the `Create VNF Identifier` request, as part of `Instantiate LCM` operation, the VNF package is on-boarded from the NFVO, if package is not already on-boarded.

  Automatic package download is controlled by the `packageDownload` configuration parameter. By default this parameter is set to `YES`.

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py read --app_ser →
ver_address localhost:8080 --name=packageDownload
YES
```

  To disable automatic package download, and to manually on-board VNF package, set `packageDownload` to `NO`, as follows:

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py update --app_s →
erver_address localhost:8080 --name=packageDownload --value=NO
NO
```

Configuration Parameters

The following configuration parameters are applicable for VNF on-board package structure in VNF-LCM:

Table 69 Package On-Board Configuration Parameters

Parameter	Default Value	Description
`vnfDescriptorBaseDirectory`	`/vnflcm-ext/current/vnf_package_repo`	Directory where parent `yaml` files are placed.
`vnfDescriptorEnvFileDirectory`	`Resources/EnvironmentFiles`	Directory where `env` files are placed.
`vnfDescriptorAdditionalFilesDirectory`	`Resources/HotFiles`	Directory where nested files are placed.
`vnfDescriptorConfigFilesDirectory`	`Resources/UserConfigurationFiles`	Directory where specific user configuration required for execution is placed.
`vnfDescriptorWrapperDirectory`	`Resources/VnfdWrapperFiles/`	Directory where Wrapper file required for workflow execution over orvnfm interfaces is placed.
`vnfDescriptorToscaMetadataDirectory`	`TOSCA-Metadata`	Directory where `tosca` metadata file is placed. This is required for `Tosca CSAR` package only.
`packageDownload`	`YES`	Automatically download the packages on VNF-LCM. This parameter is ignored when NFVO is not deployed.

`Example`

To read the value of the on-boarded package base directory, execute the command as follows:

```
[root@vnflaf-services ~]# /ericsson/pib-scripts/etc/config.py read --app_server_ →
address localhost:8080 --name=vnfDescriptorBaseDirectory /vnflcm-ext/current/vnf →
_package_repo
```

— RELATED INFORMATION —

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.14 On-boarding VNF Packages to vCloud Director Using On-board VNF Packages Workflow

VNF-LCM supports on-boarding of OVF (VNF software) packages to vCloud Director using `ERICvnfonboard_CXP345676` workflow bundle.

12.14.1 Prerequisites

- The OVF package (ovf file, vmdk images) to be on-boarded must already be in the VNF-LCM file system.
- A vCD VIM (with right user privileges) must be configured in VNF-LCM.
- vCD must be prepared with the following resources for uploading OVF package
  - A catalog for storing the vApp application template and media files.

12.14.2 On-boarding VNF Package to vCD Using Workflow

The `On-board VNF Packages` workflow performs the following steps:

1. Validates inputs provided, and if interactive is enabled, shows `User Task` forms to collect the input, if invalid.
2. Authenticates the VIM provided by the user.
3. Uses and validates the provided OVF Package as input. If not found, and interactive is enabled, the workflow lists, in the `User Task` form, all available OVF files to choose.
4. Uses the Catalog where Vapp Templates are to be created. If not found, and interactive is enabled, the workflow queries all catalogs available under the provided Organization. It lists these in the `User Task` form.
5. Creates vApp template in the selected catalog, and uploads the OVF and referenced VMDK files.

Table 70 Inputs for On-Board VNF Packages Workflow

Parameter Name	Type	Description
vAppTemplateName	String	Mandatory. Name of the vApp template to be created.
vAppTemplate description	String	Optional. Description of the vApp template.
catalogName	String	Optional. The catalog under which the vApp template has to be created.
ovfPackage	String	Optional. The OVF descriptor file to on-board.
ovfDirectory	String	Mandatory. The path where the OVF package exists in VNF-LCM file system.
VIM connection INFO **	String	Mandatory if following parameters are not provided, and interactive is false. JSON formatted string of VIM connection Info.
VIM name *	String	Optional. The name of vCD VIM where packages has to be on-boarded.
Organization Name *	String	Optional. The organization name that the catalogs are retrieved from, and where the vapp template is created.
Organization Id *	String	Optional. Organization id to retrieve the catalogs list.
Cloud username *	String	Optional. Username of the cloud to authenticate.
Cloud Password *	String	Optional. Password of the cloud.
Cloud Authurl *	String	Optional. Authurl of the vCD cloud.

Parameter Name	Type	Description
Cloud Ro-Api version	String	Optional. Cloud API version to be used. If not provided, the configured value from VNF-LCM is taken.
timeout	int	Optional. Maximum wait period in seconds for the workflow to wait until upload completes.

** VIM Connection Info JSON which holds the complete connectivity details for the VIM. If provided, then all the * parameters are optional.

* Optional. If workflow runs in interactive mode User can choose from the User Task form; otherwise mandatory.

Sample VIM Connection Info

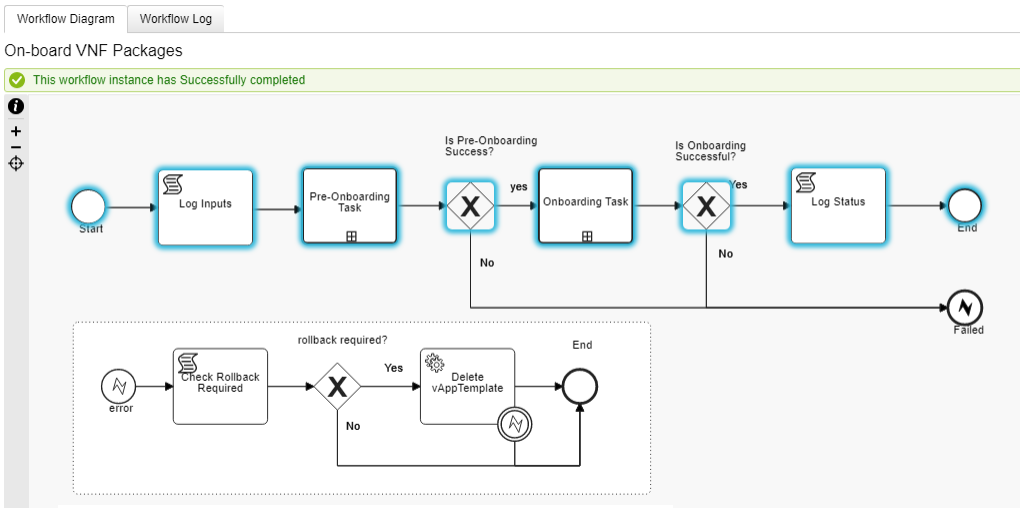
```
[{
     "id":"vim_uuid",
     "vimId":"vim-name",
     "vimType":"vimType",
     "interfaceInfo":{
        "identityEndPoint":"identity-endPoint-url"
     },
     "accessInfo":{
        "projectId":"projectId",
        "projectName":"projectName",
        "domainId":"domainId",
        "domainName":"domainName",
        "userDomain":"userDomain",
        "credentials":{
           "username":"userName",
           "password":"password" \\base64 encoded.
        }
     }
}]
```

On Successful execution of this workflow:

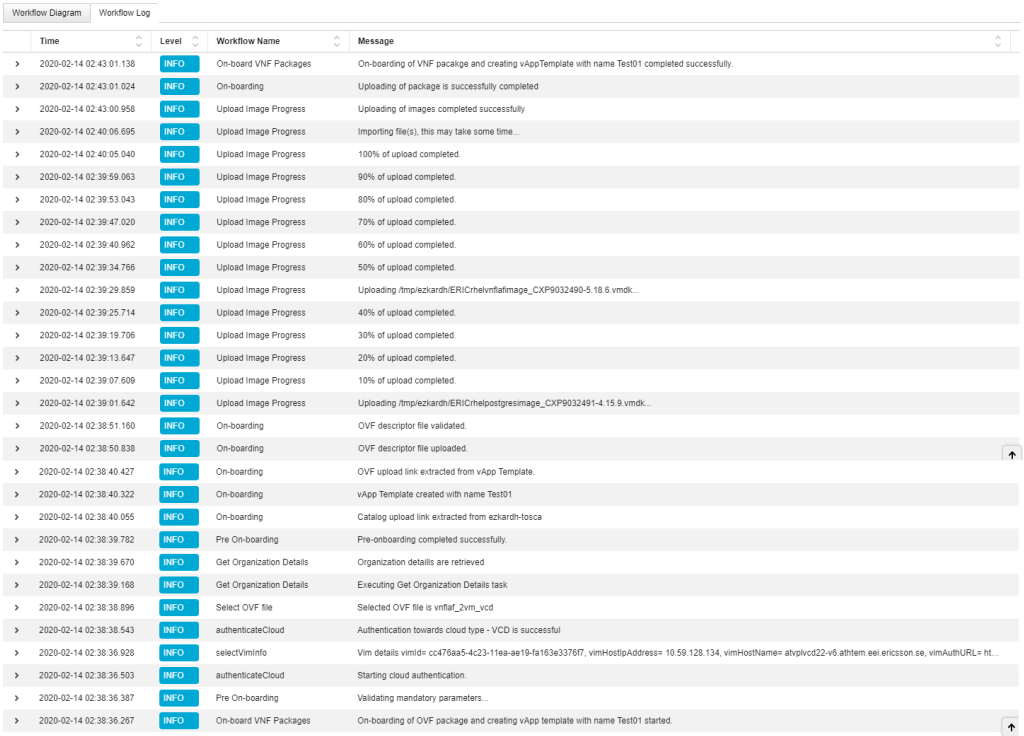
— A vApp template will be created under the selected catalog in vCloud Director

— VMs will be created in powered-off under the vApp depending on the number image files provided.

In case of failure, rollback would be performed and the vApp template would be deleted from vCloud Director.



Workflow Log Showing the Flow of Execution



Workflow Diagram | Workflow Log

Time	Level	Workflow Name	Message
2020-02-14 02:43:01.138	INFO	On-board VNF Packages	On-boarding of VNF pacakge and creating vAppTemplate with name Test01 completed successfully.
2020-02-14 02:43:01.024	INFO	On-boarding	Uploading of package is successfully completed
2020-02-14 02:43:00.958	INFO	Upload Image Progress	Uploading of images completed successfully
2020-02-14 02:40:06.695	INFO	Upload Image Progress	Importing file(s), this may take some time...
2020-02-14 02:40:05.040	INFO	Upload Image Progress	100% of upload completed.
2020-02-14 02:39:59.063	INFO	Upload Image Progress	90% of upload completed.
2020-02-14 02:39:53.043	INFO	Upload Image Progress	80% of upload completed.
2020-02-14 02:39:47.020	INFO	Upload Image Progress	70% of upload completed.
2020-02-14 02:39:40.962	INFO	Upload Image Progress	60% of upload completed.
2020-02-14 02:39:34.766	INFO	Upload Image Progress	50% of upload completed.
2020-02-14 02:39:29.859	INFO	Upload Image Progress	Uploading /tmp/ezkardh/ERICrhelvnflafimage_CXP9032490-5.18.6.vmdk...
2020-02-14 02:39:25.714	INFO	Upload Image Progress	40% of upload completed.
2020-02-14 02:39:19.706	INFO	Upload Image Progress	30% of upload completed.
2020-02-14 02:39:13.647	INFO	Upload Image Progress	20% of upload completed.
2020-02-14 02:39:07.609	INFO	Upload Image Progress	10% of upload completed.
2020-02-14 02:39:01.642	INFO	Upload Image Progress	Uploading /tmp/ezkardh/ERICrhelpostgresimage_CXP9032491-4.15.9.vmdk...
2020-02-14 02:38:51.160	INFO	On-boarding	OVF descriptor file validated.
2020-02-14 02:38:50.838	INFO	On-boarding	OVF descriptor file uploaded.
2020-02-14 02:38:40.427	INFO	On-boarding	OVF upload link extracted from vApp Template.
2020-02-14 02:38:40.322	INFO	On-boarding	vApp Template created with name Test01
2020-02-14 02:38:40.055	INFO	On-boarding	Catalog upload link extracted from ezkardh-tosca
2020-02-14 02:38:39.782	INFO	Pre On-boarding	Pre-onboarding completed successfully.
2020-02-14 02:38:39.670	INFO	Get Organization Details	Organization detaills are retrieved
2020-02-14 02:38:39.168	INFO	Get Organization Details	Executing Get Organization Details task
2020-02-14 02:38:38.896	INFO	Select OVF file	Selected OVF file is vnflaf_2vm_vcd
2020-02-14 02:38:38.543	INFO	authenticateCloud	Authentication towards cloud type - VCD is successful
2020-02-14 02:38:36.928	INFO	selectVimInfo	Vim details vimId= cc476aa5-4c23-11ea-ae19-fa163e3376f7, vimHostIpAddress= 10.59.128.134, vimHostName= atvplvcd22-v6.athtem.eei.ericsson.se, vimAuthURL= ht...
2020-02-14 02:38:36.503	INFO	authenticateCloud	Starting cloud authentication.
2020-02-14 02:38:36.387	INFO	Pre On-boarding	Validating mandatory parameters...
2020-02-14 02:38:36.267	INFO	On-board VNF Packages	On-boarding of OVF package and creating vApp template with name Test01 started.

Note:

1. vnfonboard workflow rpm, ERICvnfonboard_CXP345676, is bundled and installed as part of vCD media by default.
2. vnfonboard workflow is available to install under `/opt/ericsson/ERICvnflafsgservice_CXP9032544/conf/jboss` if a CEE or Openstack deployment. User can install the workflow rpm using `wfmgr` utility tool.
3. Workflow war file, `vnfonboard-workflows-war`, is under `/ericsson/3pp/jboss/standalone/deployments`.

12.15 Virtual Machine Live Migration using VSphere Client

Live migration of running virtual machines from one physical host server to another, with zero downtime, continuous service availability, and complete transaction integrity.

Prerequisites

1. vCloudDirector environment and vSphere Web Client with system administrator access.
2. A successful installation of the VNF-LCM vApp.

Steps

1. Login into the VCD environment with the appropriate role.
2. Go into the Vapp where the VMs are to be migrated to.
3. Right-click on the service/DB VM that is to be migrated.
4. Select **Open in vSphere Web Client**.
5. Login to the Vsphere Web client using admin credentials.
6. Click on **Actions** in the right hand panel.
7. Select **Migrate** and click **Yes** to continue.
8. In the configuration window,
    a. On the **Select the migration type** page, select **Change compute resource only** and click **Next**.
    b. On the **Select a compute resource** page,under **Hosts**, select the host to which the VM is to be migrated. Click **Next**.
    c. On the **Select network** page, select **Destination Network** for all the current networks. Click **Next**.

d. On the **Select vMotion priority** page, select **Schedule vMotion with high priority**. Click **Next**.

e. On the **Ready to complete** page, click **Finish**.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.16 Multiple Interface Configurations and Custom Routing

VNF-LCM deployment can support up to two external interfaces by default. Media contains separate Heat Templates for Single and Double interface support.

Users who want to configure three or more external interfaces must follow this procedure.

Note: For a VNF-LCM HA deployment, use the VIP of the services VM to access the system. Similarly, use the internal VIP for the DB VM. Do this on both master and standby VMs to avoid inconsistencies.

Prerequisites

— VNF-LCM is deployed successfully.

— User is able to SSH to services VM using the existing external IP.

— External network is created and User has permission to create port on it.

Steps

1. Add a new External IP to Services VM through either Openstack dashboard or CLI using the following:

   OS Dashboard:

   a. Log in to the dashboard.

   b. Select the appropriate project from the drop-down menu at the top left.

   c. On the Project tab, open the Compute tab, then click **Instances** category.

   d. Select an instance.

   e. In the menu list, from the actions column, select **Attach Interface**.

   f. Select **External network for interface** from the drop-down menu.

   g. Click **Attach Interface**.

OS CLI:

```
# openstack server add port <server> <port>

<server> Server to add the port to (name or ID)
<port> Port to add to the server (name or ID)
```

Note: This command is available from OpenStackClient 3.10.0 (pike) onwards only.

If the previous Openstack command is not available, use the `nova` command to attach the interface:

```
# nova interface-attach <Name or ID of server> --port-id <port_id>
```

Example:

```
# nova interface-attach 30870610-463e-4d29-9d66-ac4ee5c38f2e --port-id ef3ac →
e31-67fc-4d34-8d33-8d5bc7ff4891/
```

To create a port, run the following command.

```
# openstack port create <Name of this port> --network <Network ID or Name> - →
-fixed-ip ip-address=<ip-address> --security-group <security-group> --allowe →
d-address ip-address=0.0.0.0/0 --allowed-address ip-address=::/0
```

For more information, run the help command:

```
# openstack port create --help
```

Example:

```
# os port create vnflcm-ext-port --network 491522c2-f023-4162-b097-55a9a7197 →
2f3 --fixed-ip ip-address=131.160.162.105 --security-group vnflcm-sg --allow →
ed-address ip-address=0.0.0.0/0 --allowed-address ip-address=::/0
+-----------------------+------------------------------------------------- →
------------------+
| Field                 | Value                                            →
                  |
+-----------------------+------------------------------------------------- →
------------------+
| admin_state_up        | UP                                               →
                  |
| allowed_address_pairs | ip_address='0.0.0.0/0', mac_address='fa:16:3e:b0:c →
4:da'             |
|                       | ip_address='::/0', mac_address='fa:16:3e:b0:c4:da' →
                  |
| binding_host_id       |                                                  →
                  |
| binding_profile       |                                                  →
                  |
| binding_vif_details   |                                                  →
                  |
| binding_vif_type      | unbound                                          →
                  |
| binding_vnic_type     | normal                                           →
                  |
| created_at            | 2018-10-16T15:08:06                              →
                  |
| description           |                                                  →
                  |
```

```
| device_id             |                                                                                →
| device_owner          |                                                                                →
| dns_assignment        | None                                                                           →
| dns_name              | None                                                                           →
| extra_dhcp_opts       |                                                                                →
| fixed_ips             | ip_address='131.160.162.105',                                                  →
|                       | subnet_id='af93f728-7fb3-4144-8a52-23fe4ef5e117'                               →
| id                    | ef3ace31-67fc-4d34-8d33-8d5bc7ff4891                                           →
| ip_address            | None                                                                           →
| mac_address           | fa:16:3e:b0:c4:da                                                              →
| name                  | test-port                                                                      →
| network_id            | 491522c2-f023-4162-b097-55a9a71972f3                                           →
| option_name           | None                                                                           →
| option_value          | None                                                                           →
| port_security_enabled | True                                                                           →
| project_id            | cab32f669c18404d8bed0fae6bf088aa                                               →
| qos_policy_id         | None                                                                           →
| revision_number       | None                                                                           →
| security_groups       | 5e6e4a9f-af7b-445c-9573-8fddabeb8788                                           →
| status                | DOWN                                                                           →
| subnet_id             | None                                                                           →
| updated_at            | 2018-10-16T15:08:06                                                            →
+-----------------------+------------------------------------------------------------------------
------------------+
```

2. Update `host.properties` with new interface details.

   Note: The instruction in the following steps is based on following assumptions:

   — Services VM has two external interfaces already configured.

   — The interface to be added has following example network properties:

Network Properties	Value
Cidr	131.160.163.176/28
Gateway	131.160.163.177
IP Address	131.160.163.184

   Example `/ericsson/vnflcm/data/host.properties` with two example interface details:

host.properties with 2 Interface Details

```
vlans=internal,external,external2
internal_eth_interface=eth0
external_eth_interface=eth1
external2_eth_interface=eth2
external_subnet_cidr=131.160.162.0/25
external_ip_address=131.160.162.43
external_gateway=131.160.162.1
external_mtu=1500
external2_subnet_cidr=10.224.42.128/25
external2_ip_address=10.224.42.166
external2_gateway=10.224.42.129
external2_mtu=1500
```

Here, eth1 and eth2 are external network interfaces.

Update host.properties with third example interface details as listed.

host.properties with 3 Interface Details

vlans=internal,external,external2,**external3**

internal_eth_interface=eth0

external_eth_interface=eth1

external2_eth_interface=eth2

external3_eth_interface=eth3

external_subnet_cidr=131.160.162.0/25

external_ip_address=131.160.162.43

external_gateway=131.160.162.1

external_mtu=1500

external2_subnet_cidr=10.224.42.128/25

external2_ip_address=10.224.42.166

external2_gateway=10.224.42.129

external2_mtu=1500

external3_subnet_cidr=131.160.163.176/28

external3_ip_address=131.160.163.184

external3_gateway=131.160.163.177

external3_mtu=1500

Here, eth3 is the new external interface.

3. Configure interfaces in services VM.

   Run following commands in services VM:

```
# sed -i "s/service network restart/#service network restart/;s/^   configur →
e_name_resolution/\   #configure_name_resolution/" /opt/ericsson/ERICvnflafi →
```

```
nst_CXP9032542/lib/configure_network.lib
# /opt/ericsson/ERICvnflafinst_CXP9032542/bin/configure_network.sh
```

4. Perform a hard reboot on the services VM to complete the configuration.

```
# os server reboot --hard --wait <server>

<server>    Server (name or ID)
```

   Wait until reboot is finished.

5. Ping all the three external IPs and ensure that all IPs are reachable.

```
# ping -c 1 131.160.162.43
PING 131.160.162.43 (131.160.162.43) 56(84) bytes of data.
64 bytes from 131.160.162.43: icmp_seq=1 ttl=61 time=0.187 ms

--- 131.160.162.43 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.187/0.187/0.187/0.000 ms


# ping -c 1 10.224.42.166
PING 10.224.42.166 (10.224.42.166) 56(84) bytes of data.
64 bytes from 10.224.42.166: icmp_seq=1 ttl=61 time=0.273 ms

--- 10.224.42.166 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.273/0.273/0.273/0.000 ms


# ping -c 1 131.160.163.184
PING 131.160.163.184 (131.160.163.184) 56(84) bytes of data.
64 bytes from 131.160.163.184: icmp_seq=1 ttl=61 time=0.276 ms

--- 131.160.163.184 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.276/0.276/0.276/0.000 ms
```

6. Configure iptables rules for new interfaces.

```
# /opt/ericsson/ERICvnflafinst_CXP9032542/bin/configure_security.sh
```

7. Configure custom routing rule for outside networks (OSS/VIM):

   Run following command in VNF-LCM services VM:

```
#  echo "<external network or IP>  via  <gatThis routing rule means that all →
 the communications towards 141.137.207.115 is through eth1 interface.eway I →
P of interface> dev <interface name>" >> /etc/sysconfig/network-scripts/rout →
e-<interface name>
```

   Example

```
echo "141.137.207.115 via 131.160.162.1 dev eth1" >> /etc/sys →
config/network-scripts/route-eth1
```

   Where:

   — 141.137.207.115 is OSS master IP.

— 131.160.162.1 is eth1 gateway IP.

This routing rule means that all the communications towards 141.137.207.115 is through eth1 interface.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.17 Role Based Access Control (RBAC) for VNF-LCM Workflow

Role Based Access Control (RBAC) is a way to restrict access to different resources for authenticated users.

Permissions to perform certain operations are assigned to specific roles. Since users are not assigned permissions directly, management of individual user rights becomes a matter of simply assigning appropriate roles to the user's account. Role management helps in managing authorization, which enables the Security Administrator to specify the resources that users are allowed to access.

ENM has a concept of roles which define what a user can do in the system by an ENM application. There are system roles and application-specific roles which apply to a single application. There are role aliases which group various roles. It is also possible to create custom roles to define more specific access rights.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.17.1 Enable RBAC Feature

By default the RBAC feature is disabled. To use the RBAC feature it has to be enabled. To enable RBAC feature follow the steps given in ENM Details to VNF-LCM on page 223.

12.17.2 Add ENM Details

To use the RBAC feature, ENM Details are required to get the roles details from ENM. Follow the steps in Add ENM Details to VNF-LCM Framework on page 224 to add ENM Details.

12.17.3 Add Custom Roles

User access to execute workflow is managed by roles assigned to user. Roles needs to be created in ENM Role Management GUI. Please follow the ENM GUI online help to create roles for RBAC.

For creating VNF-LCM RBAC specific roles please refer to *VNF-LCM Roles* in ENM Identity and Access Management System Administrator Guide [4].

12.18 Workflow Failure Error Events

An error event will be generated in case workflow execution does not reach to `COMPLETED` state. Error event will be displayed on the ENM FM GUI.

—— RELATED INFORMATION ——

12.19 Connect to a VNF-LCM Virtual Machine on page 287

12.18.1 Add ENM Details

To send alarms ENM Details are required. Follow the steps given in ENM Details to VNF-LCM on page 223 to add ENM Details. ENM user requires `FM_Event_Administrator` role to send the error events to ENM.

12.18.2 Enable Alarm Event Communication to ENM Fault Management

To receive alarms in ENM FM GU, alarm event communications must be enabled. To enable alarm event communications follow the steps given in *Enable and Disable Event Communication To Fault Management From ENIQ* in the ENM Monitoring System Administrator Guide [22].

12.18.3 Media Upgrade from any Pre-ENM 18.06 Version (< VNF-LCM 17.17) to Higher Version

> When upgrading the media from any pre-ENM 18.06 version (< VNF-LCM 17.17) to a higher version, the password length in `vimtenants` and `vimsubtenants` table is not updated. In versions prior to 17.17 the password length is `32`. The password length changed in 17.17 to `4096`. There is no upgrade script to update the password length.

Update the password length manually by executing the following steps :

Steps

1. Login to vnflaf-db, then switch to the root user.
2. Go to directory `/opt/ericsson/ERICvnflafpersistencedb_CXP9032663/upgrade/vnflcm`:

```
cd /opt/ericsson/ERICvnflafpersistencedb_CXP9032663/upgrade/vnflcm
```

3. Check the password length in `vimtenants` and `vimsubtenants` table by running the following shell script:

```
vi getPasswordSize.sh
```

4. Insert the following into the script file:

```
#!/bin/bash

declare SUDO=/usr/bin/sudo
declare PSQL=/usr/bin/psql
declare DB=vnflafdb
declare VNFLAFDB_LOG_DIR=/var/log/vnflcm
declare VNFLAFDB_INSTALL_DIR=/opt/ericsson/ERICvnflafpersistencedb_CXP903266 →
3
SQL_PATH=${VNFLAFDB_INSTALL_DIR}/sql
VNFLAFDB_LOG_FILE=${VNFLAFDB_LOG_DIR}/vnflafdb.log

$SUDO -u postgres $PSQL --d $DB -c "select column_name,character_maximum_len →
gth from information_schema.columns where table_name = 'vimtenants' and colu →
mn_name = 'password';"
```

Note: For `vimsubtenants`, replace the value of `vimtenants` with `vimsubtenants` in table_name.

5. Change the permissions for the script file by running the following command:

```
chmod 777 getPasswordSize.sh
```

6. Run the script file:

```
./getPasswordSize.sh
```

7. If the password length is **32** then proceed to Step 8 otherwise no further action is required.
8. Create a shell script file:

```
vi updatePasswordSize.sh
```

9. Insert the following into the script file:

```
#!/bin/bash

declare SUDO=/usr/bin/sudo
declare PSQL=/usr/bin/psql
```

```
declare DB=vnflafdb
declare VNFLAFDB_LOG_DIR=/var/log/vnflcm
declare VNFLAFDB_INSTALL_DIR=/opt/ericsson/ERICvnflafpersistencedb_CXP903266 →
3
SQL_PATH=${VNFLAFDB_INSTALL_DIR}/sql
VNFLAFDB_LOG_FILE=${VNFLAFDB_LOG_DIR}/vnflafdb.log

$SUDO -u postgres $PSQL --d $DB -c "ALTER TABLE vimtenants ALTER column pass →
word TYPE character varying(4096);" >> $VNFLAFDB_LOG_FILE 2>&1

$SUDO -u postgres $PSQL --d $DB -c "ALTER TABLE vimsubtenants ALTER column p →
assword TYPE character varying(4096);" >> $VNFLAFDB_LOG_FILE 2>&1
```

10. Change the permissions for the script file by running the following command:

```
chmod 777 updatePasswordSize.sh
```

11. Run the script file:

```
./updatePasswordSize.sh
```

Note: Running the script will update the password length **4096** in `vimtenants` and `vimsubtenants` table.

To confirm run:

```
./getPasswordSize.sh
```

12.19 Connect to a VNF-LCM Virtual Machine

How to connect to the relevant VNFLCM VMs in both HA and Non-HA deployments.

Prerequisites

— Access to a Linux system with network connectivity to the VNF-LCM external network.

— This document describes details for GNU/Linux environment CLI only.

— Access to the keypair file used to Install the VNF-LCM system.

— VNF-LCM SED file.

— Value of variable external_ipv4_vip_for_services_vm or external_ipv6_vip_for_services_vm updated in SED in case of HA deployment.

— Be aware that VNF-LCM DB does not have external interface and connectivity only works over VM console or via VNF-LCM Services VM.

Steps

1. If VM is accessible over external IP address use key-pair file from your client machine VNF-LCM VM, as shown in the following example where <VM external IP> is the external IP address associated to the variable external_ipv4/6_for_vm or external_ipv4_vip_for_services_vm or external_ipv6_vip_for_services_vm

   ```
   # ssh -i <key_name>.pem cloud-user@<VM external IP>
   [local ~]$ ssh -i /var/tmp/<key_name>.pem cloud-user@<VM external IP>
   ```

2. If VNF-LCM installation successful in cloud infrastructure and VM is not accessible over external IP's, log in to respective VM using cloud infrastructures dashboard VM console.

   Note: If deployment is in OpenStack: Logon to Horizon dashboard. Click the Orchestration tab on the dashboard and choose the specific stack you want to check, identify VNF-LCM VM and open VM console.

   If deployment is in CEE: Logon to Atlas dashboard. Click the Orchestration tab on the dashboard and choose the specific stack you want to check, identify VNF-LCM VM and open VM console.

   If deployment is in VCD: Logon to VCD dashboard. Select vApps from MyCloud on the dashboard and choose the specific vApp and open console of specific VM in Virtual Machines tab.

   Switch from cloud-user to root user in VNF-LCM VMs using sudo.

   Example:

   ```
   sudo -i
   ```

Results

You are connected via SSH or console of VM. If the result is not as expected, contact local Ericsson support.

12.20 LCM Forward Compatibility Matrix

Following table shows the actions needed to enable LCM operations via workflows executed in VNFLCM for VNFs that have been instantiated with VNFLCM / workflows in previous releases.

On the vertical axis, the status by the time the VNF was instantiated is shown. On the horizontal axis, the target scenario is shown.

For example, for a VNF that was instantiated with "**VNFLCM DB without scaling info and VNF-LCM with Discovery interface**" and needs to be operated with

"**VNFLCM DB with scaling info and VNF-LCM with Discovery interface without support to update scaling info** (Release: Post R1CH)" the actions are:

— Termination / Upgrade: No action needed

— Scaling: Not supported

Table 71 LCM Forward Compatibility Matrix

Current Workflow and VNF-LCM --> ////////////// ////////////// HOW VNF WAS INSTANTIATED	Without workflows	Pre-VNF-LCM DB and No Discovery Interface(1)	VNF-LCM DB without scaling info and VNF-LCM with Discovery interface(1)	VNF-LCM DB with scaling info and VNF-LCM with Discovery interface(1) without support to update scaling info (Release: Post R1CH(2))	VNF-LCM DB with scaling info and VNF-LCM with Discovery interface(1) with support to update scaling info (Release: Post R1CJ(3))
Without workflows	No action needed	Not supported	Define VNF in VNF-LCM(4)	— Termination / Upgrade: Define VNF in VNF-LCM(4) — Scaling: Not supported	Define VNF (including scaling info) in VNF-LCM(4)
Pre-VNFLCM DB and No Discovery interface(1)	N/A	No action needed	Define VNF in VNF-LCM(4)	— Termination / Upgrade: Define VNF in VNF-LCM(4) — Scaling: Not supported	Define VNF (including scaling info) in VNF-LCM(4)

Current Workflow and VNF-LCM --> ////////////// ////////////// HOW VNF WAS INSTANTIATED	Without workflows	Pre-VNF-LCM DB and No Discovery Interface[1]	VNF-LCM DB without scaling info and VNF-LCM with Discovery interface[1]	VNF-LCM DB with scaling info and VNF-LCM with Discovery interface[1] without support to update scaling info (Release: Post R1CH[2])	VNF-LCM DB with scaling info and VNF-LCM with Discovery interface[1] with support to update scaling info (Release: Post R1CJ[3])
VNFLCM DB without scaling info and VNF-LCM with Discovery interface[1]	N/A	N/A	No action needed	— Termination / Upgrade: No action needed — Scaling: Not supported	Update scale info for VNF in VNF-LCM[4]
VNFLCM DB with scaling info and VNF-LCM with Discovery interface[1] without support to update scaling info (Release: Post R1CH[2])	N/A	N/A	N/A	No action needed	No action needed
VNFLCM DB with scaling info and VNF-LCM with Discovery interface[1] with support to	N/A	N/A	N/A	N/A	No action needed

Current Workflow and VNF-LCM --> //////////// //////////// HOW VNF WAS INSTANTIATED	Without workflows	Pre-VNF-LCM DB and No Discovery Interface(1)	VNF-LCM DB without scaling info and VNF-LCM with Discovery interface(1)	VNF-LCM DB with scaling info and VNF-LCM with Discovery interface(1) without support to update scaling info (Release: Post R1CH(2))	VNF-LCM DB with scaling info and VNF-LCM with Discovery interface(1) with support to update scaling info (Release: Post R1CJ(3))
update scaling info (Release: Post R1CJ(3))					

(1) Discovery interface - VNF-LCM CLI with a possibility to add any existing VNF to VNF-LCM.
(2) R1CH - ENM Sprint 19.02
(3) R1CJ - ENM Sprint 19.03
(4) Refer to section *Add Existing VNF to VNF-LCM and ENM Topology*.

—— RELATED INFORMATION ——

12.7.8.1 Discover VNFs in VNF-LCM and Add them to OSS-RC/ENM on page 159

12.21 Cloud Infrastructure Upgrade for v3

Cloud infrastructure upgrade to v3 without impacting VNF-LCM.

Prerequisites

— A healthy VNF-LCM

— Access to root user.

Steps

1. If a non-HA VNF-LCM deployment, then stop the external monitoring agent by removing the cron entries from `/var/spool/cron/root`:

Note: For cron entries see Configure Self-monitoring Service on page 146.

2. If an HA VNF-LCM deployment, then disable `vnflcm autorecovery`:

   Log in to the service VM using `<external_ipv4/6_vip_for_services>`, then execute the following command:

   ```
   vnflcm autorecovery disable
   ```

3. Log on to the client machine and set the environment with keystonerc file.

4. Collect the server Ids by executing the following command:

   ```
   openstack server list | grep -i <deployment_id>
   ```

   Example

   ```
   openstack server list | grep -i dep001
   ```

5. Shutdown VNF-LCM by executing the following command for those server Ids collected in Step 4 .

   ```
   openstack server stop <server id>
   ```

   Example

   ```
   openstack server stop 27caa88c-a46e-9055-3ed4-5tgg685t90y6
   ```

6. Proceed with the infrastructure upgrade.

7. Post-infrastructure upgrade, start the servers by executing the following commands:

   ```
   openstack server start <server id>
   ```

   Example

   ```
   openstack server start 27caa88c-a46e-9055-3ed4-5tgg685t90y6
   ```

   Note: `<server id>` was collected in Step 4.

8. Update the `vim url` by completing Update VIM on page 158.

9. Upgrade the external monitoring agent with a version compatible with v3 infrastructure.

   Update details of vim in `vim_and_stack_details.json`

10. If a non-HA VNF-LCM deployment, start the external monitoring agent to monitor the VNF-LCM.

    Note: To start the external monitoring agent see Configure Self-monitoring Service on page 146.

11. If an HA VNF-LCM deployment, enable `vnflcm autorecovery`:

    Log in to the service VM using `<external_ipv4/6_vip_for_services>`, then switch to root user.

    Execute the following command to enable auto-recovery.

    ```
    vnflcm autorecovery enable
    ```

13 TransportCIM Administration Tasks

When implementing common Transport functions, like link and connectivity management, ENM must deal with several node types and node versions, each one presenting its own model (e.g. ECIM, IETF or Enterprise model, Proprietary etc.). For this reason, a common TransporCIM normalized model of the NE has been introduced in ENM, so that connectivity applications can refer in a unique way to relevant MOs like ports, links etc. ENM automatically manages the population of such MOs for the NE types supporting TransportCIM Normalization.

This section contains CLI commands to verify the normalization status of a node, and to list relevant MOs normalized object values.

13.1 Browse TransportCIM Normalized Model via CLI

In order to display the tree structure of the TransportCIM normalized model the following CLI commands are used:

Prerequisites

Basic understanding of node model TransportCIM normalization concept.

Steps

1. Get child MOs of a given MO using the following command:

   ```
   cmedit describe <MOname> --namespace=OSS_TCIM
   ```

2. Get attributes of a given MO using the following command:

   ```
   cmedit describe <MOname>.* --namespace=OSS_TCIM
   ```

   The root MO of the transportcim normalized model is named “Network”: use it as starting MO name for browsing the model.

   Navigate in the model tree using the MOs discovered (i.e. listed) as a result of the previous steps.

13.2 Describe the Interfaces

The following CLI command shows the model and attributes of the interface:

```
cmedit describe interface.* --namespace=OSS_TCIM
```

13.3 Get Nodes in the Network

The following CLI command lists the nodes currently present in the Network. The term Network, when used in the context of the TransportCIM normalization service, is the domain that groups normalizable nodes only . The command lists both the nodes that are normalized, and nodes in the CREATED status (normalizable but not yet normalized). The command does not list nodes that cannot be normalized, also if these node are configured in ENM.

```
cmedit get * Network.network-id==1,*
```

13.4 Verify Normalization Status of a Node

The following CLI command verifies if a node is in NORMALIZED status:

```
cmedit get Network=1,Node=<your node>
```

The reported status can be CREATED or NORMALIZED. CREATED means that the node is associated to the Network, but the normalization process is not yet complete.

13.5 GET Interfaces of a Single Node - Top Level

The following CLI command lists all top level Interfaces of a node:

```
cmedit get * Node.node-id==<your node>,*
```

13.6 GET All Interfaces of a Single Node

The following CLI command lists all Interfaces of a node:

```
cmedit get * Node.node-id==<your node>,Interfaces.interfaces-id==1,*
```

13.7 GET All Interfaces for All Nodes

The following CLI commands list all Interfaces of nodes:

To get Interfaces for all nodes:

```
cmedit get * Node.node-id==*,Interfaces.interfaces-id==1,Interface.name==*
```

To get all Ethernet interfaces for all nodes:

```
cmedit get * Node.node-id==*,Interfaces.interfaces-id==1,Interface.name==*,Ether →
net.ethernet-id==*
```

13.8 GET All Attributes of an Interface

The following CLI command lists all attributes of an Interface:

```
cmedit get Network=1,Node=<your node>,Interfaces=1,Interface=<your Interface nam →
e>
```

13.9 Configuring PIB Parameters for TransportCIM Service

The TransportCIM PIB parameter is set to enable or disable the TranportCIM normalization processing.

13.9.1 Set the PIB Parameter to Enable the TransportCIM Normalization Service.

Prerequisites

— Understanding of node model normalization concept.

— Able to access and configure ENM service groups.

Steps

1. Follow the instructions in *Connect to a Virtual Machine*, then log on to the ipsmserv VMs.
2. As a root user give the following command in the `/opt/ericsson/PlatformIntegrationBridge/etc/` folder:

```
config.py update --app_server_address=svc-3-ipsmserv:8080 --name=autoNormali →
zationEnabled --value=true
```

13.9.2 Set the PIB Parameter to Disable the TransportCIM Normalization Service

Prerequisites

— Understanding of node model normalization concept.

— Able to access and configure ENM service groups.

Steps

1. Follow the instructions in *Connect to a Virtual Machine*, then log on to the ipsmserv VMs.

```
ssh -i /root/.ssh/vm_private_key cloud-user@svc-3-ipsmserv
```

2. As a root user, give the following command in the `/opt/ericsson/PlatformIntegrationBridge/etc/` folder:

```
config.py update --app_server_address=svc-3-ipsmserv:8080 --name=autoNormali →
zationEnabled --value=false
```

Reference List

[1] *OSS RC 14B Client Installation Instructions document update for ENM*, *21/1531-APR 901 0127*

Section: 8 Setting up AMOS and EM Application in OSS-RC

[2] *Citrix Presentation Server for UNIX Administrator's Guide*

Chapter 5: Publishing Applications and Desktops

[3] *ENM Security System Administrator Guide*,2/1543-aom9010151 Uen

[4] *ENM Identity and Access Management System Administrator Guide*, 2/1543-aom9010151-1 Uen

[5] *ENM Network Security Configuration System Administrator Guide*2/1543-aom9010151-2 Uen

[6] *ENM Public Key Infrastructure System Administrator Guide*2/1543-aom9010151-3 Uen

[7] *User Administration (CLI)*, 112/19080-cra2500056/1 Uen

[8] *Operator Access Handling*, 22/1543-axb25017 Uen

[9] *Configuring IP-Based Interfaces*, 36/1543-axb25017 Uen

[10] *ENM Product Description*, 1/1551-AOM 901 151

[11] *AMOS, Advanced MO Scripting, User Guide*, 6/1553-apr9010253 Uen

[12] *ENM Installation Instructions*, (Available from local Ericsson Support)

[13] *FLARE and Firmware Handling guide for HP/EMC*, (Available from local Ericsson Support)

[14] *ENM Site Engineering Document*, (Available from local Ericsson Support)

[15] *ENM Backup and Restore System Administrator Guide*, 3/1543-AOM 901 151

[16] *ENM Library Typographic Conventions*, 10/1551-AOM 901131

[17] *OSS Configuration for ENIQ Statistics*, (Available from Ericsson Network IQ Statistics CPI Library)

[18] *ENM Upgrade Instructions*, (Available from local Ericsson Support)

[19] *ENM Troubleshooting Guide*, 1/15901-AOM 901 151

[20] *ENM System Administrator Guide*, 1/1543-AOM 901 151

[21] *ENM Configuration System Administrator Guide*, 1/1543-AOM 901 151-1

[22] *ENM Monitoring System Administrator Guide*, 1/1543-AOM 901 151-2

[23] *ENM Performance Management System Administrator Guide*, 1/1543-AOM 901 151-3

[24] *Performance Management Description*, 3/1551-hsc10550/1 Uen

[25] *Manage Performance User Guide*, 14/1553-lza7016014/1 Uen

[26] *UE Tracer Technical Product Description*, 54/22102-axb25005/8-v2

[27] *EPG System Administrator Guide*, 30/1543-cra 119 2158-v1 Uen

[28] *Dynamic CM Import/Export Interwork Description*, 3/15519-CNA 403 2977

[29] *ENM Parameter List*, 1/190 59-AOM 901 151
[30] *ENM Node Hardening Guidelines and Instructions*, 1/174 73-AOM 901 151
[31] *ENM System Monitor User Guide*, 1/1553-CNA 403 3115
[32] *Installing Core Network Operations Manager*, Available from CNOM CPI EN/LZN 704 0220
[33] *Small Integrated ENM System Administration Guide*, 1/1543 CAN 403 3456
[34] *ENM on Cloud Backup and Restore System Administrator Guide*, 5/1543-AOM 901151
[35] *ENM Operators Guide*, 1/1553-AOM 901 151
[36] *ENM Privacy User Guide*, 2/1553-AOM 901 151
[37] *Installing UDC Dashboard*, Available from UDC Dashboard CPI EN/LZN 702 0489, available from local Ericsson support
[38] *VNF Lifecycle Management Upgrade Instructions* 1/153 72-CNA 403 331
[39] *ENM on Cloud Upgrade Instructions* 2/153 72-AOM 901 151
[40] *ENM Configuration Troubleshooting Guide*, 1/159 01-AOM 901 151-1
[41] *Flexible PM Statistics User Guide* 1/1553-HSD 101 02/1
[42] *SNMP User Guide MINI-LINK 6352* 8/1553-HRA 901 17/7