

# SAPC Troubleshooting Guide

Ericsson Service-Aware Policy Controller

## TROUBLESHOOTING

**Copyright**

© Ericsson España, S.A., 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Troubleshooting Introduction</b>	<b>1</b>
<b>2</b>	<b>Troubleshooting Tools</b>	<b>3</b>
2.1	forall	3
2.2	immHelper	4
2.3	amfHelper	4
2.4	sapcHealthCheck	4
2.5	auto_provisioning	5
2.6	BackupFormatter	6
2.7	session-handler	6
<b>3</b>	<b>Troubleshooting Functions</b>	<b>15</b>
3.1	Linux Consoles	15
3.2	Internal Database Command Line Tools	15
3.3	COM CLI	15
3.4	System Health Check	15
3.5	Processor Load (CPU and Memory)	16
3.6	Alarms and Notifications	17
3.7	Logging	17
3.8	Measurements	17
3.9	Core Files	17
3.10	System Messages	17
3.11	SAPC Reboot	17
3.12	Processor Lock and Unlock	18
<b>4</b>	<b>Troubleshooting Procedure</b>	<b>19</b>
<b>5</b>	<b>Common Faulty Situations</b>	<b>21</b>
5.1	General Failures	21
5.1.1	License Is Not Active	21
5.1.2	Processor Is Out of Service	21
5.1.3	Load Regulation	21
5.2	Provisioning Failures	22
5.3	Failures during the Initial Configuration and Provisioning in Cloud	23
5.4	Fair Usage Reporting Failures	23



5.5	Multi-Access Failures	24
5.5.1	Diameter Connection Problems	24
5.5.2	Diameter Failures	26
5.5.2.1	DIAMETER RFC 6733 Messages Failures	26
5.5.3	Gx Failures	27
5.5.3.1	General Failures	27
5.5.3.2	Gx Access Control Failures	27
5.5.3.3	Gx QoS Control for the Default Bearer and the APN Failures	28
5.5.3.4	Gx Usage Reporting Failures	29
5.5.4	Rx Failures	29
5.5.5	Sy Failures	30
5.5.6	Smp Failures	30
5.6	End User Notifications Failures	30
5.7	External Database Failures	31
5.8	EBM Failures	32
5.9	SOAP Notification Interface Failures	33
5.10	SC Absence Feature	34



# 1 Troubleshooting Introduction

The purpose of this document is to provide detailed instructions to locate and fix different problems in the SAPC typically in live sites.

This document requires strong knowledge of the product and used Component Based Architecture (CBA) components. It is addressed to both Ericsson personnel and System Administrators.

This document does not contain periodic maintenance tasks and instructions to change the configuration of the main functions within the SAPC. The [System Administrator Guide](#) contains this type of information.





## 2 Troubleshooting Tools

This section describes the tools that can be used to troubleshoot the SAPC.

### 2.1 forall

This command script launches the same CLI command or commands to several SAPC nodes, according to the following use:

```
sapcadmin@SC-1:~> forall
```

The `forall` command runs '`<command>`' in all nodes included in `<node_group>`.

Usage: `forall <node_group> '<command>'`

The valid values for `<node_group>` are the following:

- Output of the '`immHelper ng`' command
- `AllNodes`
- `SCs`
- `PLs`
- `cluster`
- `control`
- `payload`

'`<command>`' must be quoted. It can be any command that is executable through `ssh`.

#### Examples:

```
forall payload 'ps -fe | grep pcrf-proc'  
forall SCs 'hostname ; exportfs'  
forall cluster 'hostname ; uptime'  
forall PLs 'hostname; netstat -anp | grep -c 3868'
```

The reserved '`control`', '`payload`', and '`cluster`' values have the same effect as the '`SCs`', '`PLs`', and '`AllNodes`' node groups. However, they can be used even when Information Model Management (IMM) is not available because they extract the information from the cluster file system hierarchy.



## 2.2 immHelper

This command helps to know the current SAPC components state, according to the following use:

```
sapcadmin@SC-1:~> immHelper
```

```
Usage: immHelper <command: su|su2|sg|sg2|si|si2|ng|ng2> [FILTER]
```

command:

```
ng: SAPC node groups | ng2: detailed ng (show [L]ocked/[U]nlocked nodes)
su: service units    | su2: detailed su (includes node name)
sg: service groups   | sg2: detailed sg (includes node group)
si: service instances | si2: detailed si (includes availability)
comp: components
sw: installed software
```

You could grep results with 'SAPC' or whatever ...

## 2.3 amfHelper

This command executes actions on Service Units, according to the following use:

```
sapcadmin@SC-1:~> amfHelper
```

Wrapper to execute actions on Service units using amf-adm. It encapsulates the complexity of "Preinstantiable" and handles lock/unlock and lock-in/unlock-in in correct way. The repair option tries to unlock or repair matched service group and service unit that are locked or in a wrong status.

```
Use: amfHelper -f <filter> [-a <action>] [-v]
```

Parameters:

```
-f <filter>: Service unit and service group egrep filter. For example: amfH
-a <action>: stop | start | restart | status | repair . Interactive menu if
-v: verbose mode.
```

Examples:

```
amfHelper -f 'pcrf|CDiameter' -a stop
amfHelper -f 'sapc' -a stop
amfHelper -a repair -f 'sapc'
```

## 2.4 sapcHealthCheck

This command performs several checkups to verify the status of the system: SAPC deploy, TIPC communication, DRBD devices, CMW status, active FM alarms,





existing coredumps, Data Base and error logs in the system. It also provides an overall status in function of checkups results.

According to the activity (installation, upgrade, O&M or scaling workflow), this script applies different checkups and uses different criteria for overall status. Moreover, the command allows getting each checkup independently.

The use of the command is done according to the following usages:

```
sapcadmin@SC-1:~> sapcHealthCheck -h
```

```
Usage: sapcHealthCheck [-t <seconds>] [-p CHECKUP ]
       sapcHealthCheck [-t <seconds>] [BATCH]
```

#### OPTIONS:

-h, --help	help
-t, --timeout	timeout seconds for checking platform commands. Set on 3
-p, --param	specific checkup

#### CHECKUP:

```
SAPCInstallation
Connectivity
DRBD
CoreMiddleware
Alarms
CoreDumps
SystemOperative
DataBase
ErrorLogs
```

#### BATCH

-d, --deploy	deployment/installation checkups
-u, --upgrade	upgrade checkups
-o, --oam	operation and maintenance checkups
-s, --scaling	scaling checkups

---

---

### Attention!

This command reports Health Check NOK for Not ACTIVE zone in Geographical Redundancy deployments.

---

---

## 2.5

### auto\_provisioning

This command is used to automatically provision data, during the SAPC deployment, according to the following use:



```
sapcadmin@SC-1:~> auto_provision
```

Usage:

```
/usr/local/bin/auto_provision start
```

This command will provision the data included in provided files REST /cluster/storage/no-backup/auto\_provision/initial\_provisioning.rest

## 2.6 BackupFormatter

BackupFormatter tool is used to export information contained in backups of SAPC internal database.

The BackupFormatter is distributed in the `sapc_toolkit_cxp9035521_<version>.tar.gz` package. This package can be downloaded from Ericsson Software Gateway under a unique SAPC ticket number together with SAPC software. Refer to the Release Notes document to check the version and the ticket number.

Follow these steps in order to use the BackupFormatter from a machine external to the SAPC.

1. EXT\_MACHINE> `mkdir sapc_toolkit`
2. EXT\_MACHINE> `tar -xf sapc_toolkit_cxp9035521_<version>.tar.gz -C sapc_toolkit`
3. EXT\_MACHINE> `cd sapc_toolkit/BackupFormatter/bin`
4. EXT\_MACHINE> `./BackupFormatter`

Usage: `./BackupFormatter <backup directory> <command> [<arguments>]`

For more information about BackupFormatter, refer to [Database Backup Formatter User Guide](#).

## 2.7 session-handler

This tool accesses Database Service (DBS) and retrieves the session model for the subscriber identified by a specific traffic identifier (MSISDN, IMSI or SIP-URI). The tool shows relevant information regarding the bound sessions of the subscriber for all the applicable protocols (Gx, Rx, Sy, and Sd), unless explicitly stated to restrict this information to AF sessions (Rx protocol) only.

Depending on the requested action, the tool also allows the deletion of retrieved sessions from the DBS. By default, any request for a “delete” action requires confirmation from the user.



Session deletion is not done in DBS directly. The pcrf-proc is notified about those Gx or Rx sessions, or both, which need to be deleted from the subscriber's session model, so it can send the appropriate termination messages to them. As a result, all their bound sessions will be deleted from the DBS.

By default, the verification that the required sessions are deleted from DBS is configured to be done with a maximum of five checking retries and an interval of two seconds between those retries. These default values can be modified in the configuration file that is described later.

If, at the end of this retries, the deletion of some session could not be verified by any reason, a descriptive warning message is shown and the unverified sessions are listed.

This tool runs in any traffic payload, according to the following use:

```
sapcadmin@PL-3:~> sudo session-handler --help
```

```
Usage: session-handler --trafficId <trafficId> --action <action>
[--onlyAF] [--noConfirm] [--cfg <configFile>] | --help
```

Allowed options:

<b>--help</b>	Help message.
<b>--trafficId arg</b>	The traffic ID value (IMSI, MSISDN, or SIP-URI value) identifying the subscriber whose sessions are going to be shown or deleted.
<b>--action arg</b>	Action to be executed. The valid values are: show, delete.
<b>--onlyAF</b>	A restriction to apply the requested action only to AF sessions (Rx).
<b>--noConfirm</b>	A command which forbids asking for confirmation in the case of a delete action request.
<b>--cfg arg</b>	A specific configuration file to replace the default configuration file and to override the default configuration values.

The session-handler tool uses a configuration file to set internal values. By default, the tool uses the `session-handler.cfg` file that can be found in the following path:

```
"/storage/system/config/sapc/session-handler.cfg"
```

**Note:** A different configuration file can also be provided to the tool by means of the `--cfg` option.

The default content in this configuration file is as follows:



```
# thrift configuration to connect with pcrf-proc
pcrf-proc-thrift-host=localhost
pcrf-proc-thrift-port=${@SAPC_PORT_PCRF@}
thrift-connection-timeout=30
thrift-send-recv-timeout=200
# maximum number of retries while checking for sessions deletion
deletion-verification-retries-number=5
# number of seconds between retries while checking for sessions deletion
deletion-verification-time-between-retries=2
```

The value of <SAPC\_PORT\_PCRF> is established by the deployment of the SAPC. It corresponds to the port used by the pcrf-proc process to receive session termination requests through the Thrift protocol.

The first example shows the session model for the subscriber identified by the “34600001401” trafficId:

```
sapcadmin@PL-3:~> sudo session-handler --trafficId
34600001401 --action show
```

```
Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [34600001401] -> adminSubsId [admin_subscriber_1]
-----
Session data model corresponding to trafficId -> [34600001401]
-----
Gx sessions:
  IP session (Gx):
    ipAddr (IP@APN@PCEF) =
[224.236.110.211@APN2@ggsnNodeHostname.nodeHostRealm.com]
    diamSessionId (Gx) =
[tc_14_01_Rx_Sy_simpleCase;ggsnNodeHostname.nodeHostRealm.com;2;2960140]
    creationTime = [May 07, 2018; 15:06:17]
    modificationTime = [May 07, 2018; 15:06:31]
    peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
    pccRules = { [Chat], [dyn_unknownService] }

Af sessions:
  Af session (Rx):
    afSessionId (IP@diamSessionId) =
[224.236.110.211@tc_14_01_Rx_Sy_simpleCase;afNodeHostname.nodeHostRealm.com;2;5]
    creationTime = [May 07, 2018; 15:06:31]
    peerId = [afNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
    state = [0]

Sy sessions:
  Sy session:
    sySessionId (diamSessionId) =
```



```
[sapcOwnHostId.operatorRealm.com;1525451758940294;288230377677163504;3460000
```

Sd sessions:

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

The second example shows only the AF sessions (Rx protocol) for the subscriber identified by the “34600001401” trafficId:

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 3460
0001401 --action show --onlyAF
```

```
Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [34600001401] -> adminSubsId [admin_subscriber_1]
-----
Session data model corresponding to trafficId -> [34600001401]
-----
```

Af sessions:

```
Af session (Rx):
  afSessionId (IP@diamSessionId) =
[224.236.110.211@tc_14_01_Rx_Sy_simpleCase;afNodeHostname.nodeHostRealm.com;
  creationTime = [May 07, 2018; 15:06:31]
  peerId = [afNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
  state = [0]
```

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

The third example shows how to delete only the AF sessions (Rx protocol) for the subscriber identified by the “34600001401” trafficId, after the required confirmation:

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 3460
0001401 --action delete --onlyAF
```

```
Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [34600001401] -> adminSubsId [admin_subscriber_1]
-----
Session data model corresponding to trafficId -> [34600001401]
-----
```





```
modificationTime = [May 07, 2018; 17:27:10]
peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
pccRules = { [Chat] }
```

Af sessions:

Sy sessions:

```
Sy session:
sySessionId (diamSessionId) = [sapcOwnHostId.operatorRealm.com;1525
451758940294;288230377677163505;34600001401]
```

Sd sessions:

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # #
```

The fourth example shows how to delete the whole session model for the subscriber identified by the “34600001401” trafficId, without requiring any confirmation from the user:

```
sapcadmin@PL-3:~> sudo session-handler --trafficId 346000
01401 --action delete --noConfirm
```

Initializing DBN API

Waiting for DBN (false)...

```
INFO: trafficId [34600001401] -> adminSubsId [admin_subscriber_1]
```

```
-----
Session data model corresponding to trafficId -> [34600001401]
-----
```

Gx sessions:

```
IP session (Gx):
ipAddr (IP@APN@PCEF) =
[98.169.218.38@APN2@ggsnNodeHostname.nodeHostRealm.com]
diamSessionId (Gx) =
[tc_14_01_Rx_Sy_simpleCase;ggsnNodeHostname.nodeHostRealm.com;2;6078370]
creationTime = [May 07, 2018; 17:48:07]
modificationTime = [May 07, 2018; 17:48:21]
peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
pccRules = { [Chat], [dyn_unknownService] }
```

Af sessions:

```
Af session (Rx):
afSessionId (IP@diamSessionId) =
[98.169.218.38@tc_14_01_Rx_Sy_simpleCase;afNodeHostname.nodeHostRealm.com;2;
creationTime = [May 07, 2018; 17:48:21]
```



```

peerId = [afNodeHostname.nodeHostRealm.com@nodeHostRealm.com]
state = [0]

Sy sessions:
Sy session:
  sySessionId (diamSessionId) = [sapcOwnHostId.operatorRealm.com;15254517589402
401]

Sd sessions:

# # # # # # # # # # # # # # # # # # # # # #

Deleted Gx session corresponding to:
  ipAddr (IP@APN@PCEF) =
[98.169.218.38@APN2@ggsnNodeHostname.nodeHostRealm.com]
  diamSessionId (Gx) =
[tc_14_01_Rx_Sy_simpleCase;ggsnNodeHostname.nodeHostRealm.com;2;6078370]
  creationTime = [May 07, 2018; 17:48:07]
  peerId = [ggsnNodeHostname.nodeHostRealm.com@nodeHostRealm.com]

Deleted Af session corresponding to:
  id (IP@diamSessionId) =
[98.169.218.38@tc_14_01_Rx_Sy_simpleCase;afNodeHostname.nodeHostRealm.com;2;16
  creationTime = [May 07, 2018; 17:48:21]
  peerId = [afNodeHostname.nodeHostRealm.com@nodeHostRealm.com]

Deleted Sy session corresponding to:
  diamSessionId (Sy) =
[sapcOwnHostId.operatorRealm.com;1525451758940294;288230377677163506;346000014
  adminSubsId = [admin_subscriber_1]

```

This output demonstrates how the tool shows the subscriber's session model with all the sessions to be deleted first (Gx, Rx and Sy). Then, without asking the user for confirmation, it removes all those sessions from the DBS.

Running the tool again to show the remaining session model for the subscriber identified by the "34600001401" trafficId, it confirms that none of the deleted sessions are there anymore:

```

sapcadmin@PL-3:~> sudo session-handler --trafficId
34600001401 --action show

```





```
Initializing DBN API
Waiting for DBN (false)...
INFO: trafficId [34600001401] -> adminSubsId [admin_subscriber_1]
-----
Session data model corresponding to trafficId -> [34600001401]
-----
Gx sessions:
```

Sy sessions:

Sd sessions:

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # #
```





## 3 Troubleshooting Functions

### 3.1 Linux Consoles

The Linux Console is accessed using the SSH protocol towards the System Controller processors using the **sapcadmin** user through the **<OAM VIP>**. For more details, refer to [System Administrator Guide](#). If the **<OAM VIP>** is unavailable, the operation and maintenance scripts cannot be used. See [Section 5.10](#) on page 34.

For more information about available commands, check the LDE Management Guide.

### 3.2 Internal Database Command Line Tools

The internal database command line tools can provide useful information about database status. These tools offer data on a cluster level, like the status of the processors that form the SAPC and some other information regarding memory consumption and internal connections. To execute every tool, use the command `clurun.sh` from any of the SC or PL processors.

```
sapcadmin@SC-1:~> clurun.sh
```

### 3.3 COM CLI

This console provides a direct CLI for the COM subsystem and also a textual representation of Management Information Model (MIM). For more details, refer to [System Administrator Guide](#).

### 3.4 System Health Check

To check if the SAPC is working properly, the `sapcHealthCheck` is used. It provides information about:

- SAPC Software components installed
- Nodes communication through TIPC
- DRBD status
- CMW and AMF status
- Alarms, coredumps, and error logs in the system
- DBS status



The next example shows the script output using default options for a succeed state:

```
sapcadmin@SC-X:~> sudo sapcHealthCheck
===== HEALTH CHECK REPORT =====

Checking the SAPC is installed...
SAPC installation --> OK --> All the 29 ERIC-SAPC SDPs installed are used [main

Checking TIPC communication...
TIPC --> OK --> All the 4 available nodes at TIPC level are up.

Checking DRBD devices...
DRBD device --> OK

Checking CMW status...
CMW status --> OK --> All the "node comp app su si sg siass csiass pm" are OK (

Checking AMF status...
AMF status --> OK --> All the AMF entities are OK.

Checking active FM alarms...
Alarms --> OK --> There are no active FM Alarms.

Checking existing coredumps...
Coredumps --> OK --> There are no coredumps.

Checking system operative...
External peers configured and in use --> OK

Checking Data Base...
All Data Base agents working normally --> OK

Checking error logs in the system...
No errors in the system --> OK

*** SAPC HEALTH CHECK SUMMARY ***
WARNINGS:  0
ERRORS:    0
*****

SAPC Health Check finished: OK
```

## 3.5 Processor Load (CPU and Memory)

The specific commands to check that CPU and Memory load are described in [Preventive Maintenance](#).



## 3.6 Alarms and Notifications

For information about how to check system alarms, refer to [Preventive Maintenance](#).

If any alarm is raised, act on the corresponding OPI to make it cease.

## 3.7 Logging

For further information, about the Logging events generated by the SAPC, refer to [Logging Events](#).

## 3.8 Measurements

The traffic measurements generated by the SAPC also provide useful information when troubleshooting a problem. For more information, refer to [Measurements](#).

## 3.9 Core Files

To check the existence of system core files, refer to [Preventive Maintenance](#).

If any, send then to the next level of maintenance support for analysis.

## 3.10 System Messages

Important information about the general status of the different processors can be found as root user in the following files found in any SC processor:

```
root@<SC-X>: /var/log/<node-id>/auth*
```

```
root@<SC-X>: /var/log/<node-id>/kernel*
```

```
root@<SC-X>: /var/log/<node-id>/messages*
```

**Note:** Where <SC-X> is SC-1 or SC-2

Where <node-id> is SC-1, SC-2, PL-3, PL-4 or PL-n

## 3.11 SAPC Reboot

The SAPC can be reloaded with the following commands.

---

---

### Caution!

The procedure implies almost 30-seconds downtime until the internal database is operating again.

---

---

1. Log on to the system with **sapcadmin** user, through <OAM VIP>.
2. Perform the reboot of the SAPC.  
  

```
sapcadmin@SC-X> sudo cmw-cluster-reboot [--yes]
```

If --yes is specified, the command does not require confirmation.
3. Wait until the node is back.
4. Log on again to the system with **sapcadmin** user, through <OAM VIP>.
5. Check the status of the node according to [Preventive Maintenance](#).

## 3.12 Processor Lock and Unlock

A processor on the SAPC can be locked from the node. A processor locked means that it is not part of the cluster, until the unlocked command is performed (the processor comes back to the node).

1. Log on to the system with **sapcadmin** user, through <OAM VIP>.
2. Perform the lock of the processor in the SAPC.

```
sapcadmin@SC-X> sudo cmw-node-lock <processor>
```

---

---

### Caution!

Traffic performance can be affected until the processor is unlocked.

---

---

3. To get the processor back on the node, execute the following command:  
  

```
sapcadmin@SC-X> sudo cmw-node-unlock <processor>
```
4. Check the status of the node according to [Preventive Maintenance](#).



## 4 Troubleshooting Procedure

Troubleshooting a problem in the SAPC requires the use of one or more functions described in the previous chapters. The correct use of these tools is needed to prevent overload situations. In a faulty situation, they must be used in the right order to ensure an efficient location of the fault:

1. Perform a System Health Check described on Section 3.4 on page 15.
2. Check processors load. See Section 3.5 on page 16.
3. Check for alarms in the system. To do that, follow Section 3.6 on page 16.
4. Check for logs in the system. To do that, follow Section 3.7 on page 17.
5. Check the traffic measurements. See Section 3.8 on page 17.
6. Check the capacity measurements and purchased capacity licenses. See Section 3.8 on page 17.
7. Check system core dumps files, follow Section 3.9 on page 17.
8. Check system messages. See Section 3.10 on page 17.

A troubleshooting workflow is shown in Figure 1.

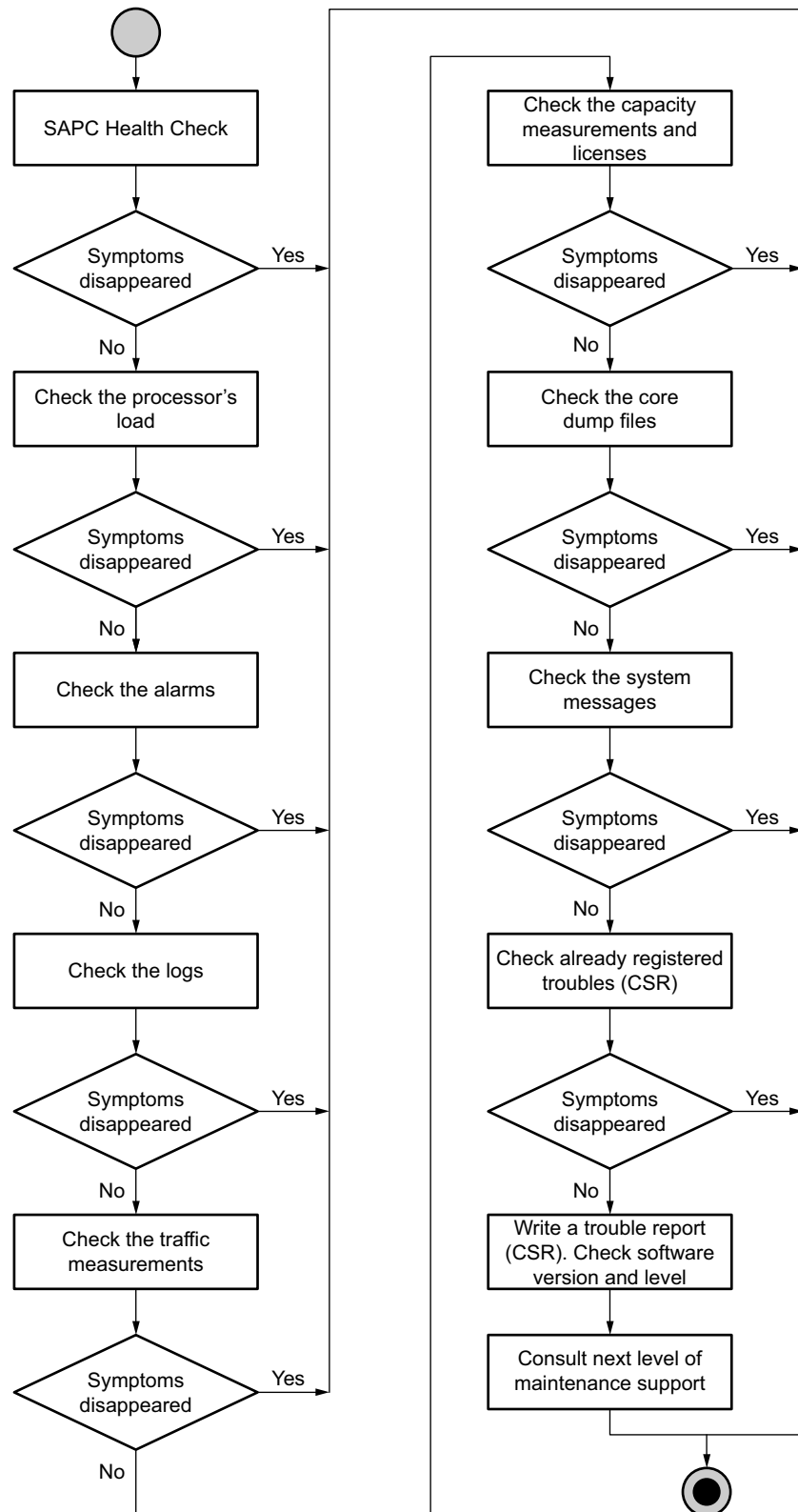


Figure 1 Troubleshooting Workflow





## 5 Common Faulty Situations

In the following sections, some common problems and possible solutions are described that can appear during normal operation.

### 5.1 General Failures

#### 5.1.1 License Is Not Active

If traffic is continuously rejected or misprocessed, it can be caused by a license which is not properly installed, expired, or whose capacity is exceeded. If a license has expired, contact supply organization to request an extension.

1. Check License Manager status and configuration. Refer to [View License Information](#).
2. Check for specific alarms regarding License Manager or capacity licenses. Refer to [License Management](#).
3. If there are active alarms about capacity licenses, check the capacity measurements (see Section 3.8 on page 17) and purchased capacity licenses.
4. If a license is not properly activated or installed, reinstall licenses. Refer to [Install License Key File](#).
5. If a license has expired, contact supply organization to request an extension.
6. To restore system functionality temporarily in extraordinary situations, activate the Emergency Unlock mode as described in [Activate Emergency Unlock Mode](#). Consider that the number of activations is limited.

#### 5.1.2 Processor Is Out of Service

The SAPC is composed of several processors. If, during operation, any processor goes out of service, the rest of the traffic processors must handle all the traffic, so it can result in a higher load situation for them. To verify and correct the situation, follow the next step:

1. Check if the SAPC platform component status is correct following the steps described in Section 3.4 on page 15.

#### 5.1.3 Load Regulation

The SAPC continuously monitors CPU usage load and memory consumption. If the values of these parameters exceed a configurable threshold, the SAPC rejects requests in Gx, Rx, and Sy interfaces, and session establishments in the Smp interface by answering with the DIAMETER\_TOO\_BUSY (3004) value in the Result-Code AVP. It can also reject REST provisioning messages (with

HTTP 503 error), SOAP notifications (with HTTP 500 error), and time triggered reauthorizations. This is to guarantee a graceful behavior of the SAPC in overload situations. If DIAMETER\_TOO\_BUSY, HTTP 503, or HTTP 500 messages are detected continuously during a prolonged period:

1. Follow the System Health Check described in Section 3.4 on page 15. If one or more nodes are not working properly, the rest of them can be in an overload situation.
2. Check the value for Load Regulation constraints as described in the [Overload Control User Guide](#) User Guide. Adjust the values to the manufacturer recommendation.
3. If none of the previous actions detected a malfunction or erroneous configuration, it is most likely that the SAPC is applying load regulation because of high resource consumption. If this situation persists over time, contact the next level of maintenance support.

## 5.2 Provisioning Failures

Representational State Transfer (REST) is used as an interface for the SAPC provisioning purpose. Through REST services commands, it is possible to provision the SAPC with Subscribers, Subscribers Groups, and Policies.

- If REST provisioning is not successful:

Verify that the provisioned information is correct according to the following documents:

- [Configuration Guides](#)
- Database Access
- Integration in Software Defined Network
- Provisioning REST API

- If the internal database does not accept more provisioning entries:

Check that the database storage capacity limit is not reached.

To verify this, launch the following command that shows the amount of used and free memory:

```
sapcadmin@SC-1:~>clurun.sh collect_stats -d dbn
```

```
Result from [PL-3.dbn]: DbsService=DBN,DbsPU=PL-3
DbsPU.VS.DBS.Mem.NormalHeap.Free 0
DbsPU.VS.DBS.Mem.NormalHeap.Used 61889
DbsPU.VS.DBS.Mem.RecordHeap.Free 3276215
DbsPU.VS.DBS.Mem.RecordHeap.PUsed 0
DbsPU.VS.DBS.Mem.RecordHeap.Used 0
DbsPU.VS.DBS.Mem.TotalHeap.Free 3276215
```



```

DbSPU.VS.DBS.Mem.TotalHeap.Used 61889
...
Result from [PL-4.dbn]: DbService=DBN,DbSPU=PL-4
DbSPU.VS.DBS.Mem.NormalHeap.Free 0
DbSPU.VS.DBS.Mem.NormalHeap.Used 61978
DbSPU.VS.DBS.Mem.RecordHeap.Free 3276215
DbSPU.VS.DBS.Mem.RecordHeap.PUsed 0
DbSPU.VS.DBS.Mem.RecordHeap.Used 0
DbSPU.VS.DBS.Mem.TotalHeap.Free 3276215
DbSPU.VS.DBS.Mem.TotalHeap.Used 61978
...

```

---

### Caution!

If there is no total heap available, contact Ericsson personnel for more information.

---

## 5.3 Failures during the Initial Configuration and Provisioning in Cloud

Optionally, the SAPC can be automatically configured and provisioned in Cloud deployments. During deployment time, the files containing the data are injected to the SAPC and the **auto\_provision** script is executed (to find more details, refer to SAPC VNF Descriptor Generator Tool).

In some scenarios, in which the Cloud Infrastructure where the SAPC is deployed presents a slow connection speed, this automatic procedure could not have been executed.

1. Confirm if the script for automatic configuration and provisioning is successfully executed, looking at the contents of the following log file in SC-1:

```
SC-1:~ # cat /var/log/auto_provision/auto_provision.log
```

2. Instead, if the last text message is "Waiting NDB nodes to be STARTED ...", the execution fails. Then, do it manually from SC-2:

```
SC-2:~ # /usr/local/bin/auto_provision
```

## 5.4 Fair Usage Reporting Failures

If no quota is received from the SAPC, verify that:

1. The subscriber or subscriber group has `usageLimits` information configured.
2. The content of the Usage Limits is syntactically right. This can be checked by parsing the JSON structure with some external tool (for example: <http://jsonformatter.curiousconcept.com>).



3. Subscription Date < Current time < Expiry Date.
4. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to TRUE.

If quota = 0 is received from the SAPC, check that the applicable Reporting Groups are enabled:

5. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to TRUE.

For further information, refer to [Configuration Guide for Fair Usage](#).

## 5.5 Multi-Access Failures

TCP connectivity exists between peer and the SAPC.

### 5.5.1 Diameter Connection Problems

If there is any failure related to Diameter traffic, verify the following checks:

1. Verify Diameter Flow Policy.

Check through an ECLI session if all Flow Policies are defined. For that purpose, execute the next command with the associated output:

```
> show ManagedElement=1,Transport=1,Evip=1,EvipAlbs=1,EvipAlb=alb_trf,EvipFlowPolicies=1
```

```
EvipFlowPolicies=1
  EvipFlowPolicy=SCTP_diameter
  EvipFlowPolicy=diameter_ipv4_<DIAMETER-PORT1>
  EvipFlowPolicy=diameter_ipv4_<DIAMETER-PORT2>
  EvipFlowPolicy=soap
```

2. Verify VIP address for traffic or Diameter port.

Check through an ECLI session if the VIP-TR and DIAMETER-PORT are set. For that purpose, execute the next command with the associated output considering the VIP address for traffic handling and the 3868-port values:

```
>show all ManagedElement=1,Transport=1,Evip=1,EvipAlbs=1,EvipAlb=alb_trf,EvipFlowPolicies=1
```

```
EvipFlowPolicies=1
  EvipFlowPolicy=SCTP_diameter
    addressFamily="ipv4"
    dest="<VIP-TR>"
    protocol="sctp"
```



```

soGrp="1011250"
usageState=ACTIVE

EvipFlowPolicy=diameter_ipv4_<DIAMETER-PORT1>
addressFamily="ipv4"
dest="<VIP-TR>"
destPort="<DIAMETER-PORT1>"
protocol="tcp"
targetPool="PLs_rr"
usageState=ACTIVE

EvipFlowPolicy=diameter_ipv4_<DIAMETER-PORT2>
addressFamily="ipv4"
dest="<VIP-TR>"
destPort="<DIAMETER-PORT2>"
protocol="tcp"
targetPool="PLs_rr"
usageState=ACTIVE

EvipFlowPolicy=soap
addressFamily="ipv4"
dest="<VIP-TR>"
destPort="8080"
protocol="tcp"
targetPool="PLs_rr"
usageState=ACTIVE

```

### 3. Diameter status processes.

Check through an SSH connection as sapcadmin user to any SC processor if the C-diameter status is OK. For that purpose, execute the next command with the associated output, considering N as the number of PL processors:

```
sapcadmin@SC-X: /> amfHelper -f CDiameter -a status
```

```
Searching SUs filtering (egrep) by 'CDiameter' ...
```

```

***>> CDiameter
[Node]      [Service Unit DN]

```

```
Done!
```

```
PL-N      safSu=PL-N,safSg=NWA,safApp=ERIC-sv.SVCDiameter
```

```
Done!
```



## 5.5.2 Diameter Failures

### 5.5.2.1 DIAMETER RFC 6733 Messages Failures

If there is any problem with the establishment of Diameter connections, it can be owing to one of the following reasons:

1. Capabilities-Exchange-Request (CER) message is received from an unknown peer.

- Check that `acceptFrom` in IMM is `<Empty>` to allow any unknown peer. This is the **recommended** configuration since it is not possible to change this value without a restart of system.

**Note:** SC-X:~ # `immlist -a host -a acceptFrom `immfind -c OtpdiaTransportTcp``

```
host=:all
acceptFrom=\<Empty\>
host=:all
acceptFrom=\<Empty\>
```

- If the value is defined, check that the neighbor node host IP matches the “PCRE” regular expression.

**Note:** Example accepts peers with IPs from 10.\* and 172.\* using PCRE expression `10.*|172.*`:

SC-1: # `immlist -a host -a acceptFrom `immfind -c OtpdiaTransportTcp``

```
host=all
acceptFrom="10.*|172.*"
host=all
acceptFrom="10.*|172.*"
```

2. Receive request for an unsupported application.

- Check Application Id and Supported Vendor Id.

`immlist -a supportedVendorId -a authApplicationId `immfind -c OtpdiaApplications``

- Check the Vendor Specific Application Id grouped AVP:

`immlist -a vendorSpecificApplicationId `immfind -c OtpdiaApplications``

```
vendorSpecificApplicationId=otpdiaVendorSpecificApplicationId=Gx
```



From this, we use vendorSpecificApplicationId=otpdiaVendorSpecificApplicationId=Gx.

```
immlist otpdiaVendorSpecificApplicationId=Gx,otpdiaProduct=
SAPC -a vendorId -a otpdiaVendorSpecificApplicationId -a
authApplicationId
```

```
vendorId=10415
```

```
otpdiaVendorSpecificApplicationId=otpdiaVendorSpecificApplicationId=G
authApplicationId=16777238
```

3. Incorrect Origin-Host, Origin-realm, Host-IP-Address from SAPC.

— Check originHost, originRealm, hostIpAddress

```
immlist `immfind -c OtpdiaService`
```

4. Not possible to establish two or more connections to the same peer

— Check that restrictConnections is set to false for otpdiaService=Pc  
rf,otpdiaProduct=SAPC in IMM

## 5.5.3 Gx Failures

### 5.5.3.1 General Failures

If any AVP related to a concrete control (Bearer Access Control, Service Access Control, QoS Control for the Default Bearer and the APN, Usage Reporting, and so on) is not obtained in CCA or RAR messages, it can be owing to one of the following reasons:

1. Check the bit for the corresponding control received in the Gx-Capability-List AVP within CCR requests.
2. Check also the corresponding control received in the Supported-Features AVP.
3. Check the configuration for the controls for the PCEF sending the Gx requests.

If all mobile session establishments are suddenly rejected, it can be caused by the number of mobile sessions exceeding the capacity license. Check for License Manager alarms and verify the mobileActiveSessions measurement against the purchased capacity license.

### 5.5.3.2 Gx Access Control Failures

To identify the value of the Diameter Result-Code AVP in the answer message. To get this value, a protocol analyzer is recommended to be used (for example, Wireshark). For further information about the Result-Code meaning, refer to Gx Interface Description.

If service authorization is not successful, it can be owing to one of the following reasons:

1. Subscriber received in the Subscription-Id AVP is not found in the SAPC and "Unknown" special EPC-Subscriber entry is not provisioned.
  - Check if the UnknownSubscribers measure is abnormally high.
  - Check if the subscriber is correctly provisioned.
2. Service authorization result is not the expected one:
  - Check if the specific service is correctly defined.
  - Check in the provisioned Subscriber profile the allowed and blacklist services.
3. For static services, check if the service is included in the applicable Rule Space (either the one indicated by the PCEF or the one decided by the SAPC).
4. Check also whether the service is within the list of services to redirect provisioned both for the subscriber and the active groups the subscriber belongs to.
5. Check policies for the specific service. To detect if there is any error in the policy evaluation, activate the warning logging level temporarily.
6. To check if there is a protocol error, activate the warning logging level temporarily.
7. The session to be updated does not exist:
  - Check as root user if there are new logs with the following message:  
  
Non-Persistent data storage is empty.  
  
In the next path:  
  
`SC-X:~ # /cluster/storage/no-backup/coremw/var/log/saflog/sapc/`  
  
— If the processors load allows it, activate the warning logging level temporarily to check if the session was previously removed.

### 5.5.3.3 Gx QoS Control for the Default Bearer and the APN Failures

If the QoS Control for the Default Bearer and the APN result are not the expected ones:

1. Verify that the Bearer QoS Control applies for the PCEF (check the configuration of the PCEF in the SAPC).





2. Check the values in the QoS-Negotiation, QoS-Upgrade, and QoS-Information AVPs.
3. Check if the gxQoSDowngraded, gxQoSUpgraded, and gxQoSDeactivated measures are abnormally high. If so, check the values configured in the QoS Profiles associated with the QoS Control for the Default Bearer and the APN (either per service or per bearer) policies. Compare them to the values received in the requested QoS Profile.

For details about provisioning and configuration, refer to *Configuration Guide for Bearer QoS Control and Bandwidth Management*.

#### 5.5.3.4

#### Gx Usage Reporting Failures

If no quota is received from the SAPC, verify that:

1. The subscriber or subscriber group has the usageLimits information configured.
2. The contents of the Usage Limits are syntactically right. This can be checked by parsing the JSON structure with some external tool (for example <http://jsonformatter.curiousconcept.com>).
3. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to “TRUE”. To detect if there is any error in the policy evaluation, activate the **info** logging level temporarily.
4. Subscription Date < Current time < Expiry Date

If no more volume quota available is received from the SAPC, check that the applicable Reporting Groups are enabled:

5. Accumulation policies for the applicable Reporting Groups (and included counters) evaluate to “TRUE”. To detect if there is any error in the policy evaluation, activate the **info** logging level temporarily.

For further information, refer to *Configuration Guide for Fair Usage*.

#### 5.5.4

#### Rx Failures

If there is any failure related to the Rx interface, verify the following:

1. Check service classification related configuration, provisioning, and policies.  
Refer to *Configuration Guide for Dynamic Policy Control (Rx)*.
2. Check service authorization related configuration, provisioning, and policies.  
Refer to *Configuration Guide for Dynamic Policy Control (Rx)*.
3. Check service qualification related configuration, provisioning, and policies.



Refer to Configuration Guide for Dynamic Policy Control (Rx).

4. Check if the counters `rxAasFailed`, `rxRaasFailed`, `rxAsasFailed`, `rxAasUnableToComply`, `rxAasInvalidInfo`, `rxAasIpSessionNotAvailable`, `RxTerminateUnknownSessions`, `rxRarsTimeout`, and `rxAsrsTimeout` measures are abnormally high.

If all AF session establishments are suddenly rejected, it can be caused by the number of AF sessions exceeding the capacity license. Check for License Manager alarms and verify the `afActiveSessions` measurement against the purchased capacity license.

### 5.5.5 Sy Failures

If there is any failure related to the Sy interface, verify the following:

1. Check the Subscriber Charging related configuration, provisioning, and policies.

Refer to Configuration Guide for Integration with OCS for Spending Limit Reporting (Sy).

2. Check if the counters `sySlrsTimeout`, `sySlasFailed`, `sySnasFailed`, `syStrsTimeout`, and `syStasFailed` measures are abnormally high.
3. If SLR message is sent by the SAPC to the Online Charging System, but STR message is not sent later on: verify that the destination realm sent within the SLR and the origin realm received within the SLA are both properly defined in the SAPC DIAMETER routing table. Refer to Configuration Guide for Integration with OCS for Spending Limit Reporting (Sy).

### 5.5.6 Smp Failures

If there is any failure related to Smp interface, verify the following:

1. Check the PDN-GW and SPID related configuration, provisioning and policies.

Refer to Configuration Guide for Mobility Based Policy Control for Overlay Deployments (Smp).

2. Check if the counters `sxCcasInitFailed`, `sxCcasInitInvalidAvp`, `sxCcasInitMissingAvp` and `sxCcasInvalidInfo` measures are abnormally high.

## 5.6 End User Notifications Failures

If the SMS/SOAP Notifications fail to be sent, verify the following:

1. Verify that the `enableDelivery` attribute of the **NotificationConfig** COM object is set to "TRUE".
2. Verify that the End User Notifications are configured properly:



- For SMS notifications:

Check if the **SMSCenter** and **SMSDestination** COM object values are correctly configured under the **Network** COM object.

- For SOAP notifications:

Check if the **WebServiceEndPoints**, **WebServiceEndPoint**, and **WsDestination** COM object values are correctly configured under the **Network** COM object.

3. Check if the notification policies are correctly configured.
4. Check if the "ConnectionNotificationServerFailed" alarm is raised.
5. Check logs for end-user notifications.

Further information on configuring end-user notifications can be found in [Configuration Guide for End User Notifications](#).

## 5.7 External Database Failures

If there is any failure related to access to external database, verify the following:

1. Check through an ECLI session if the VIP address for access to external database <VIP-ExtDB> is defined on the Abstract Load Balancer (alb\_trf).

```
> show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
EvipAlb=alb_trf, EvipVips=1
```

```
EvipVip= <VIP-ExtDB>
```

```
EvipVip= <VIP-TR>
```

2. Check through an ECLI session if the Local IP address for access to external database is defined on the Entity Data object.

```
> show ManagedElement=1, PolicyControlFunction=1, EntityData=1
```

```
localIp=<VIP-ExtDB>
```

3. Check through an ECLI session the IPs defined for external database.

```
> show ManagedElement=1, PolicyControlFunction=1, EntityData=1,
EDSources=1, EDSource=ExternalRepository
```

```
EDSource=ExternalRepository
```

```
definition="def ExternalRepository () { dataSource
= { url = \"\"; query = \"\"; } fieldDef = { ips =
\"136.225.72.9;136.225.72.17;136.225.72.25\"; port = \"389\";
} }"
```



4. Check if there is connection to any of the External databases from your PL.

```
sapcadmin@PL-X:~> ping -I <VIP-ExtDB> <External Database IP>
```

5. Check if the external database outgoing connections are correctly established within the active IP. 64 connections per PL are expected.

```
sapcadmin@SC-X:~> forall sapc.payload "hostname; lsof
-i@<External Database IP>:389 | grep -i ESTABLISHED | wc -l"
PL-3
64
PL-4
64
```

Further information on configuring access to External Database can be found in [Database Access](#).

## 5.8 EBM Failures

If there is any failure related to the Event-Based Monitoring (EBM) interface, verify the following:

1. Check through an ECLI session if the VIP address for access to the EBM Server <VIP-EBM> is defined on the Abstract Load Balancer (alb\_trf).

```
> show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
EvipAlb=alb_trf, EvipVips=1
```

```
EvipVip= <VIP-EBM>
```

```
address = <IP-EBM>
```

```
state = "ACTIVE"
```

2. Check if the enable attribute of the EventBasedMonitoring COM object is set to true.

```
>show ManagedElement=1,PolicyControlFunction=1,EventBasedMonito
ring=1,EbmBusinessEvents=1,EbmBusinessEvent=QUOTA_GRANTED
EbmBusinessEvent=QUOTA_GRANTED
    ebmServerIds
        ""
    enable=true
>
```

3. Check if at least one EBM Server is configured correctly.
4. Check if individual EBM events are enabled.



5. Check if the `EbmCommunicationFailure` or `EbmBufferOverflow` alarms are raised.
6. Check if the `ebmBusinessEventsNotSent` measure is abnormally high.

## 5.9

### SOAP Notification Interface Failures

If there is any failure related to SOAP notification interface, verify the following:

1. Check through an ECLI session the Flow Policy for SOAP incoming notification service. Misconfigured SOAP incoming notification service flow policy in eVIP prevents the correct binding of the SOAP server process to the listening port.

```
> show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
EvipAlb=alb_trf, EvipFlowPolicies=1
```

```
EvipFlowPolicy=soap
```

This is the print definition of flow policy:

```
> show ManagedElement=1, Transport=1, Evip=1, EvipAlbs=1,
EvipAlb=alb_trf, EvipFlowPolicies=1, EvipFlowPolicy=soap
```

```
EvipFlowPolicy=soap
  addressFamily="ipv4"
  dest="<VIP-ExtDB>"
  destPort="8080"
  protocol="tcp"
  targetPool="PLs_rr"
  usageState=ACTIVE
```

2. Check if the SOAP incoming notification service port 8080 is listening on Abstract Load Balancer where the VIP for access to external database is configured (alb\_tr).

```
SC-X:~ # forall sapc.payload "hostname ; netstat -nap | grep
:8080 | grep LISTEN"
```

PL-10					
tcp	0	0	10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-11					
tcp	0	0	10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-12					
tcp	0	0	10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-5					
tcp	0	0	10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-6					
tcp	0	0	10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-7					
tcp	0	0	10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-8					



tcp	0	0 10.41.30.53:8080	0.0.0.0:*	LISTEN
PL-9				
tcp	0	0 10.41.30.53:8080	0.0.0.0:*	LISTEN

Further information on configuring SOAP incoming notification web service can be found in [SOAP Notification Interface Description](#).

## 5.10 SC Absence Feature

If the **<OAM VIP>** interface is unavailable, it can be because of failures in both **SCs**. In this scenario, **OAM** features are restricted, but the traffic can be still processed for 15 more minutes before the whole cluster reboots.

1. Recover at least one of the two **SCs** to prevent the cluster from rebooting. Once it is recovered, the **<OAM VIP>** is available.
2. If no **SC** is recovered in 15 minutes, the cluster goes down. In this scenario, recover at least one of the **SCs** and the system restarts normally. If the **SC** does not recover or the system does not restart normally, contact the next level of maintenance support.