

CSCF Health Check

Call Session Control Function

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2016–2018. 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

1	Introduction	1
1.1	Prerequisites	1
2	Health Check Tasks	3
3	Automated Health Check Procedures	5
3.1	Preparations	5
3.2	Automatic Health Check	5
3.3	Automatic Health Check Verdict	6
3.4	Automatic Health Check Options	7
4	Configuration File	9
4.1	Configuration File Format	9
4.2	Configuration File Location	9
4.3	Configuration Parameters	9
5	Manual Health Check Procedure	13
5.1	Preparations	13
5.2	Check Release Information	13
5.3	Verify Status of Alarms	14
5.4	Verify Controller Status	15
5.5	Verify System Status	17
5.6	Monitor Network Connectivity	18
5.7	Verify Processor Status	20
5.8	Verify Administrative and Operational State	21
5.9	Verify Diameter Stack Status	22
5.10	Verify CPU Load and Memory Use	26
5.11	Verify eVIP Status	30
5.12	Check System Environment Variables	31
5.13	Check Availability of DNS Servers	32
5.14	Check Status of SIP Interfaces	34
5.15	Store Health Check Report	35
6	Report Problems	37
7	Example of Configuration File	39



8	Example of Automatic Health Check Results	41
9	File Management	45



1 Introduction

This document describes how to perform the health check on the Call Session Control Function (CSCF). The health check tasks described in Section 2 on page 3 are recommended to be performed before and after a system update/upgrade, a normal backup, or during the periodic maintenance. For information about health check-related Key Performance Parameters (KPIs), refer to *Check CSCF Key Performance Indicators*.

1.1 Prerequisites

This section describes the prerequisites for performing the health check procedure.

1.1.1 Documents

Before starting this procedure, ensure that the following information or documents are available:

- The CSCF Release Note corresponding to the current CSCF version.
- Node Configuration.
- IP Traffic Network Diagram.

1.1.2 Tools

The following tools are required:

- Workstation with Secure Shell (SSH) capabilities.
- A web browser on the workstation.

1.1.3 Conditions

The user performing the health check procedure must fulfill the following conditions:

- Know the CSCF system on a System administrator level.
- Know the external Virtual IP (VIP) Addresses and the System Controller (SC) address of the cluster node.



- Know the password of an Operation & Maintenance (O&M) user with sufficient rights to display CSCF Configuration Management (CM) and Fault Management (FM) parameters.
- Know the password for the troubleshooting user.
- Has authorized access to the CSCF with `sudo/root` privileges for the required troubleshooting and Operations, Administration, and Maintenance (OAM) actions.



2 Health Check Tasks

The most common health checks are covered by the automated procedure as described in Section 3 on page 5. In troubleshooting situations or when more control is desired, the checks can be performed using the manual procedures in Section 5 on page 13. The recommended periodicity for some of the most useful tasks is shown in Table 1.

Table 1 Critical Areas to Be Monitored Regularly or in Certain Situations

Section	Description
Alarms	Alarms often need to be monitored (once per hour).
Network connectivity	Can be often done (once per hour). Can be run on need basis or once per day.





3 Automated Health Check Procedures

This section describes how to perform the health check of the CSCF by running an automated script.

3.1 Preparations

This section describes the preparations required to execute the health check.

3.1.1 Log on to System Controller

Log on to system controller using SSH:

```
ssh -A <username>@<OAM IP>
```

3.1.2 Obtain Persistent Storage Area Paths

To obtain the different persistent storage area paths for the system, enter these commands on the node:

```
<configuration-path> = cmwea config-location-get
```

```
<storage-path> = cmwea no-backup-location-get
```

3.2 Automatic Health Check

Automatic health check includes the following checks:

- Release Information
- Current alarms
- Controller status
- CPU load
- Diameter ports listening
- Administrative and operational state
- Processor outage
- Network connectivity
- Memory use



- Evolved Virtual Internet Protocol (eVIP) status
- System status
- System environment variables
- Performance indicators

The results are printed to the console and a report is created, which requires manual verification.

3.2.1 Run Automatic Health Check

To run the automatic health check:

1. Run the health check script:

```
CscfHealthCheck
```

Note: The first time the script is run, or when there are mandatory parameters in the configuration file that are missing a value, the user is prompted to enter values for these. For more information about the configuration file and configuration parameters, see Section 4 on page 9.

2. Check the results printed to the console. An example of the results is shown in Section 8 on page 41.
3. Find the generated report files with the most recent date and time in the directory:

```
<storage-path>/vcscf_cxp9034345/healthcheck/reports/CscfHealthCheckReport<nodename>_<timestamp>.html
```

```
<storage-path>/vcscf_cxp9034345/healthcheck/reports/PM_INDICATORS_Report_<nodename>_<timestamp>.html
```

Health Check report files can be fetched using File Management. For more information, see Section 9 on page 45.

3.3 Automatic Health Check Verdict

The verdict is a way to inform the user of the status of the individual checks. The definitions of the different verdicts are shown in Table 2.

Table 2 Verdict Definitions

Verdict	Description
INFO	Information for the user, not checked by the script.



Verdict	Description
OK	Task passed.
VERIFY	Manual verification needed.
FAIL	Problem detected by the script.
ERROR	An error occurred, script update needed or system broken.

3.4 Automatic Health Check Options

The behavior of the health check script can be further customized by providing command line options. Supported options are listed in Table 3.

Table 3 Command Line Options Supported By Health Check Script

Option	Meaning
-h, --help	Print a brief help message and exit.
-r REPORT, --report REPORT	Specify the location to save report. The default is <code><storage-path>/vcscf_cxp9034345/healthcheck/reports</code>
-f FILENAME,, --filename FILENAME	Filename prefix for generated files. The default is <code>CscfHealthCheckReport_<nodename></code> and <code>PM_INDICATORS_Report_<nodename></code>
-q, quiet	Prints only the verdict for each check in console.
--cpu-max=CPU_MAX	Sets the threshold, in %, that the CPU load must reach for the healthcheck script to flag VERIFY instead of OK.





4 Configuration File

This section describes the configuration file used by the automatic health check script. It contains parameters that can be configured by the user.

4.1 Configuration File Format

Each setting consists of a line with the format `key=value`. Multiple-value settings are handled by including multiple lines with the same key.

Lines that start with `#` are ignored.

4.2 Configuration File Location

The configuration file is stored in `<config-path>/vcscf_cxp9034345/healthcheck/`, and is called `<user-name>.config`.

The file permission is set to `read` and `write` for the user.

4.3 Configuration Parameters

If the configuration file does not exist, the script creates a default configuration file including default values. Some parameters are mandatory but are not given any default value. For the following parameters, the user is prompted to enter a value. The configuration file is updated with the following entered values:

- `Cluster User`
- `Cluster Password`
- `O&M Host`
- `O&M User`
- `O&M Password`

4.3.1 Cluster Port

The parameter `cluster.port` configures the port used when SSH to system controller on the cluster.

This parameter is mandatory. By default this parameter has the value 22.



4.3.2 Cluster User

The parameter `cluster.user` configures the user used when SSH to system controller on the cluster.

This parameter is mandatory. By default this parameter is not configured and must be updated for the script to work properly.

4.3.3 Cluster Password

The parameter `cluster.password` configures the password used when SSH to system controller on the cluster.

This parameter is mandatory.

Note: For security reasons, the password is not stored in the configuration file, and the password cannot be read from the configuration file. As a result, the user is always prompted for password.

4.3.4 Granularity Period

The parameter `granularity.period` configures the `cscfHealthCheck` script to select the Performance Management (PM) log files with the specific Granularity Period. The value of this parameter is in seconds. The default value for Granularity Period is 300 seconds.

Note: It is recommended to configure PM jobs with single Granularity Period.

4.3.5 O&M Host

The parameter `oam.host` configures the host address or hostname to be used when SSH to Ericsson Command-Line Interface (ECLI).

This parameter is mandatory. By default this parameter is not configured and must be updated for the script to work properly.

Note: If the O&M Movable IP (MIP) is configured on the system, the O&M host address is automatically retrieved and not prompted for user input.

4.3.6 O&M Port

The parameter `oam.ecliport` configures the port used when SSH to ECLI.

This parameter is mandatory. By default this parameter has the value 2022.

4.3.7 O&M User

The parameter `oam.user` configures the user used when SSH to ECLI.



This user must have a role that allows the user to log on to ECLI and display alarms and configuration parameters.

This parameter is mandatory. By default this parameter is not configured and must be updated for the script to work properly.

4.3.8 O&M Password

The parameter `oam.password` configures the password to be used when O&M user SSH to ECLI.

This parameter is mandatory.

Note: For security reasons, the password is not stored in the configuration file, and the password cannot be read from the configuration file. As a result, the user is always prompted for password.

4.3.9 Maximum CPU Load

The parameter `cpu.max` sets the threshold that the CPU load must reach for the `healthcheck` script to flag `VERIFY` instead of `OK`.

The default value is **81%**.

4.3.10 Counters

The parameter `pmf.counters` configures counters from which the `CscfHealthCheck` script retrieves information.

By default, the following counters are configured:

- `cscfAcceptedRegistrations`
- `cscfExpiredRegistrations`
- `cscfRejectedRegistrations`
- `cscfFailedSessions`
- `cscfScscfAssignments`
- `cscfCxSelPullinitRegistrations`
- `cscfCxPullUnableToComplys`
- `cscfACABackup`
- `cscfNBASuccess`
- `cscfSipDigestAuthenticationSuccess`
- `scscfGibaSuccess`





5 Manual Health Check Procedure

This section describes the procedure for manually checking the health of the system for the CSCF.

5.1 Preparations

This section describes the required preparations performed before checking the node health.

5.1.1 Log on to System Controller

Log on to the system controller using SSH:

```
ssh -A <username>@<OAM IP>
```

5.2 Check Release Information

For more information regarding the ECLI, refer to *Ericsson Command-Line Interface User Guide*.

To check the CSCF Release Information:

1. Log on to ECLI:

```
ssh -A <username>@<OAM IP> -p <port>
```

2. Check the release parameter in ManagedElement:

```
show ManagedElement=<nodename>
```

Example output:

```
ManagedElement=1
  managedElementType="vCSCF"
  release="1.6.0"
  CscfFunction=1
[...]
```

The following example output shows an vCSCF 1.6.0 Emergency Package (EP) release.



```
ManagedElement=1
managedElementType="vCSCF"
release="1.6.1"
CscfFunction=1
Equipment=1
SystemFunction
```

Note: The example shows the vCSCF 1.6.0 EP release. There are rare cases that an vCSCF EP is released only with a platform component update. Such EP releases do not result in an update of the release parameter in ManagedElement.

The application and platform component versions can be checked by following section *Software Level Checks* in *CSCF Troubleshooting Guideline*.

3. Check the release parameter in CscfFunction=1:

```
show ManagedElement=<nodename>, CscfFunction=1
```

Example output:

```
CscfFunction=1
release="CXP9035589/1 R7A (1.6.0-4) "
userLabel=""
CSCF-Application=CSCF
CscfDomainRoutingApplication=CscfDomainRouting
CscfEosApplication=CscfEos
DIA-CFG-Application=DIA
DNS-Application=DNS
ExtNetSel-Application=ExtNetSelection
ExtNetSel-Application=ExtNetSelection2
ICMP-Application=ICMP
LdapClientApplication=LdapClientApplication
LI-Application=LI
NumberNormalisation=NumberNormalisation
SigComp-Subsystem=SigComp
```

5.3 Verify Status of Alarms

For more information regarding the ECLI, refer to *Ericsson Command-Line Interface User Guide*.

To verify the status of alarms:

1. Log on to ECLI:

```
ssh -A <username>@<OAM IP> -p <port>
```

2. Verify that there are no raised alarms:



```
show ManagedElement=<nodename>,SystemFunctions=1,Fm=1
-m FmAlarm
```

If an alarm is found, follow the procedures in the related alarm Operating Instructions.

Example output:

```
FmAlarm=65
  activeSeverity=WARNING
    additionalText="Detailed Information: Link disabled by
      OAM, IRP Cause: 14"
  eventType=COMMUNICATIONSALARM
  lastEventTime="2014-04-14T15:35:35+02:00"
  majorType=193
  minorType=2250572778
  probableCause=14
  sequenceNumber=65
  source="ManagedElement=jambala, connId =conn1,
    stack=CSCFRF,Host=LABSPTOFFCHA.ericsson.se"
  specificProblem="Diameter, Link Disabled"
FmAlarm=89
  activeSeverity=WARNING
    additionalText="Detailed Information: Link disabled by
      OAM, IRP Cause: 14"
  eventType=COMMUNICATIONSALARM
  lastEventTime="2014-04-14T15:35:43+02:00"
  majorType=193
  minorType=2250572778
  probableCause=14
  sequenceNumber=89
  source="ManagedElement=jambala, connId =conn1,
    stack=CSCFRF,Host=LABSPTOFFCHA2.ericsson.se"
  specificProblem="Diameter, Link Disabled"
```

3. Log off from ECLI:

```
exit
```

5.4 Verify Controller Status

To verify the controller status:

1. Check that the system controller is connected to the other half, that is, the output is `Connected`.

Note: If not, and if it does not resolve itself within 15 minutes, contact next level of maintenance support.

```
ssh `cmw-hostname-get SC-1` drbdadm cstate all
```

```
ssh `cmw-hostname-get SC-2` drbdadm cstate all
```



Connected

2. Check that the disk state is normal state UpToDate.

Note: If not, and if it does not resolve itself within a reasonable time frame, contact next level of maintenance support.

```
ssh `cmw-hostname-get SC-1` drbdadm dstate all
```

```
ssh `cmw-hostname-get SC-2` drbdadm dstate all
```

UpToDate

3. Check if the system controller is primary or secondary.

```
ssh `cmw-hostname-get SC-1` drbdadm role all
```

```
ssh `cmw-hostname-get SC-2` drbdadm role all
```

Primary/Secondary

On the primary controller, the field starts with `Primary/`, typically the value is `Primary/Secondary`.

On the secondary controller, the field starts with `Secondary/`, typically the value is `Secondary/Primary`.

Note: If an error has occurred, then the field can contain other values.

4. Display the state from CoreMW point of view:

```
cmw-status -v siass | grep OpenSAF -A2
```

Example output:



```
safSISU=safSu=PL-3\,safSg=NoRed\,safApp=OpenSAF,
safSi=NoRed4,safApp=OpenSAF
  HASTate=ACTIVE(1)
  HAREadinessState=READY_FOR_ASSIGNMENT(1)
safSISU=safSu=SC-2\,safSg=NoRed\,safApp=OpenSAF,
safSi=NoRed1,safApp=OpenSAF
  HASTate=ACTIVE(1)
  HAREadinessState=READY_FOR_ASSIGNMENT(1)
safSISU=safSu=SC-2\,safSg=2N\,safApp=OpenSAF,safSi=SC-2N,
safApp=OpenSAF
  HASTate=STANDBY(2)
  HAREadinessState=READY_FOR_ASSIGNMENT(1)
safSISU=safSu=PL-4\,safSg=NoRed\,safApp=OpenSAF,
safSi=NoRed3,safApp=OpenSAF
  HASTate=ACTIVE(1)
  HAREadinessState=READY_FOR_ASSIGNMENT(1)
--
safSISU=safSu=SC-1\,safSg=2N\,safApp=OpenSAF,safSi=SC-2N,
safApp=OpenSAF
  HASTate=ACTIVE(1)
  HAREadinessState=READY_FOR_ASSIGNMENT(1)
safSISU=safSu=SC-1\,safSg=NoRed\,safApp=OpenSAF,
safSi=NoRed2,safApp=OpenSAF
  HASTate=ACTIVE(1)
  HAREadinessState=READY_FOR_ASSIGNMENT(1)
```

5. Verify that HASTate is ACTIVE or STANDBY for:

```
safSISU=safSu=SC-1\,safSg=2N\,safApp=OpenSAF,safSi=SC-2N,safApp=OpenSAF
safSISU=safSu=SC-2\,safSg=2N\,safApp=OpenSAF,safSi=SC-2N,safApp=OpenSAF
```

5.5 Verify System Status

To verify the system status:

1. Display the vDicos Middleware (MW) status:

```
immllist lpmsv=LPMSvSite
```

For more information, refer to *vDicos Management*.

Example output:



Name	Type	Value(s)
=====		
lpmsvState	SA_STRING_T	Idle
lpmsv	SA_NAME_T	lpmsv=LPMSvSite(15)
SaImmAttrImplementerName	SA_STRING_T	LPMSvImplementer
SaImmAttrClassName	SA_STRING_T	LPMSv
SaImmAttrAdminOwnerName	SA_STRING_T	IMMLOADER

2. Verify that lpmsvState is Idle.

5.6 Monitor Network Connectivity

To verify the network connectivity:

1. Use **ssh** to connect to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <username>@<OAM IP>
```

2. Display Container Distribution Service (CDSv) connections of the control server:

```
clurun.sh -c ctrlsrv_print_conn
```

Example output:



Result from [.cdsv.director]:

DBSv

```
Server port:          <1.1.2:3151823157>
Client port:         <1.1.2:3151823091>
Connection status:   established
Client version:      1
Server version:      1
Common version:      1
Queued messages:     0
```

LPMSv

```
Server port:          <1.1.2:3151692052>
Client port:         <1.1.2:3151757573>
Connection status:   established
Client version:      1
Server version:      1
Common version:      1
Queued messages:     0
```

Result from [SC-2.cdsv.director]:

DBSv

```
Server port:          <1.1.2:3151823157>
Client port:         <1.1.2:3151823091>
Connection status:   established
Client version:      1
Server version:      1
Common version:      1
Queued messages:     0
```

LPMSv

```
Server port:          <1.1.2:3151692052>
Client port:         <1.1.2:3151757573>
Connection status:   established
Client version:      1
Server version:      1
Common version:      1
Queued messages:     0
```

3. Verify that Connection status is established.
4. Display CDSv Connections of the distribution server:

```
clurun.sh -c distsrv_print_conn
```

Example output:



```
Result from [.cdsv.director]:

...

DBSv on safAmfNode=SC-2,safAmfCluster=myAmfCluster
  Server port:          <1.1.2:3151888694>
  Client port:          <1.1.2:3152347376>
  Connection status:    established
  Client version:       1
  Server version:       1
  Common version:       1
  Queued messages:      0

LPMSv on safAmfNode=PL-3,safAmfCluster=myAmfCluster
  Server port:          <1.1.2:3151692090>
  Client port:          <1.1.3:4076011595>
  Connection status:    established
  Client version:       1
  Server version:       1
  Common version:       1
  Queued messages:      0
```

Note: Only part of output is shown here.

5. Verify that `Connection status` is established.

5.7 Verify Processor Status

To verify the processor status:

1. Use `ssh` to connect to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <username>@<OAM IP>
```

2. Enter the command:

```
cdsv-get-user-state
```

Example output:



```
Result from [.cdsv.user.director.DBSv]:
DBSv - cluster state: Idle
Agent server port: <1.1.2:3151692031>, listening on 92345,99
Agents (count: 4):
Agent[0x670430] node: safAmfNode=PL-3,safAmfCluster=myAmfCluster,
  port: <1.1.3:4075946061>
Idle: yes, Halting: no
Number of operation states: 0
...

Result from [.cdsv.user.director.LPMSv]:
LPMSv - cluster state: Idle
Agent server port: <1.1.2:3151692041>, listening on 12345,1
Agents (count: 4):
Agent[0x7099e0] node: safAmfNode=PL-3,safAmfCluster=myAmfCluster,
  port: <1.1.3:4075946057>
Idle: yes, Halting: no
Number of operation states: 0
...
```

Note: Only part of output is shown here.

3. Verify that DBSv - cluster state and LPMSv - cluster state are both Idle.

5.8 Verify Administrative and Operational State

For more information regarding the ECLI, refer to *Ericsson Command-Line Interface User Guide*.

To verify that the CSCF is ready to handle traffic:

1. Log on to ECLI:

```
ssh -A <username>@<OAM IP> -p <port>
```

2. Display parameter cscfISPOperationalState and cscfAdministrativeState under CSCF-Application=CSCF:

```
show ManagedElement=<nodename>,CscfFunction=1,CSCFApplication=CSCF
```

Example output:



```

CSCF-Application=CSCF
  cscfActiveUserMethod
    ""
  cscfAdministrativeState=UNLOCKED
  cscfCXDestinationHost="LAB7HSS.ericsson.se"
  cscfCXDestinationRealm="cx.ericsson.se"
  cscfCXOriginHost="LAB24CSCF.ericsson.se"
  cscfCXOriginRealm="cscf.ericsson.se"
  cscfDomainAlias
    "cscf24.lab"
  cscfDomainBasedPSIRoutingEntry
    "/^psi\\.cscf24\\.lab/i"
  cscfGlobalNumberNormalizationPhoneContext=""
  cscfISPOperationalState=ENABLED
  cscfPhoneContext="+46"
  ...

```

Note: Only part of output is shown here.

3. Verify that the parameter `cscfISPOperationalState` has the value `ENABLED` and parameter `cscfAdministrativeState` has the value `UNLOCKED`.

4. Log off from ECLI:

```
exit
```

5.9 Verify Diameter Stack Status

To verify the status of the Diameter stack:

1. Use `ssh` to connect to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <username>@<OAM IP>
```

2. Check which nodes in the cluster that are started:

```
cdsv-get-node-state -s
```

3. Check which started nodes that have `DIASharedProcPool` hosted. If already known, continue with Step 4.

```
cdsv-print-node -v | egrep -w 'DIASharedProcPool' -B 3
```

Example output:

```

Result from [.cdsv.director]:
Node[0x8ebfd0] id: 9 name: safAmfNode=PL-10,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 0100100a
  UserApp states: [ 0: 00000003 1: 00000003 ]

```



```

Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x8b9940] id: 8 name: safAmfNode=PL-11,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 0100100b
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x876600] id: 6 name: safAmfNode=PL-12,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 0100100c
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x822540] id: 3 name: safAmfNode=PL-3,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001003
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x7df040] id: 1 name: safAmfNode=PL-4,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001004
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x767b30] id: 0 name: safAmfNode=PL-5,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001005
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x8a8be0] id: 7 name: safAmfNode=PL-6,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001006
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharpProcPool ]
Node[0x8658c0] id: 5 name: safAmfNode=PL-7,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001007
  UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharpPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool

```



```
DicosDbClassPot_Pool DIASStackProcPool ]
Node[0x843f00] id: 4 name: safAmfNode=PL-8,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001008
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASStackPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASStackProcPool ]
Node[0x7eff50] id: 2 name: safAmfNode=PL-9,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001009
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASStackPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASStackProcPool ]
```

4. Log on to ECLI:

```
ssh -A <username>@<OAM IP> -p <port>
```

5. Check portNr, ipAddressesList, and sctpAddressesList for each configured diameter interface. If already known, continue with Step 8.

```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-A
pplication=DIA,DIA-CFG-StackContainer=CSCFCX,DIA-CFG-O
wnNodeConfig=CSCFCX
```

```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-A
pplication=DIA,DIA-CFG-StackContainer=CSCFRF,DIA-CFG-O
wnNodeConfig=CSCFRF
```

```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-A
pplication=DIA,DIA-CFG-StackContainer=CSCFRO,DIA-CFG-O
wnNodeConfig=CSCFRO
```

```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-A
pplication=DIA,DIA-CFG-StackContainer=CSCFRX,DIA-CFG-O
wnNodeConfig=CSCFRX
```

Example output:



```
DIA-CFG-OwnNodeConfig=CSCFCX
allowConnectFromUnknownNode=false
diaVendorId="10415"
enabled=true
firmwareRevision="0"
hostId="LAB24CSCF.ericsson.se"
ipAddressesList
    "0:10.35.38.14"
loadRegulationEnabled=false
maxNumberOfRetries="2"
maxRequestPendingTime="4"
permissions=63
portNr="3868"
productName="ISP-CSCF"
realm="cscf.ericsson.se"
sctpHandlerLogLevel="DEFAULT"
sendErrorAtOverload=false
shareTree=""
supportedAuthAppIds
    "16777216"
    "16777217"
supportedVendorsIds
    "0"
    "10415"
    "13019"
supportedVendorSpecificApps
    "0:0:16777216:16777216"
    "1:10415:16777216:16777216"
    "2:0:16777217:16777217"
    "3:10415:16777217:16777217"
traceSctpHandler="DEFAULT"
transportLayerType="1"
```

6. Make a notation of portNr, ipAddressesList, and sctpAddressessList.

7. Log off from ECLI:

```
exit
```

8. Check the Transmission Control Protocol (TCP) diameter port status and verify the ports that are available for use for each interface on each node:

```
ssh -A <node hostname> netstat -an | grep <tcp
address>:<port>
```

Example output:

```
tcp      0      0 10.35.38.14:3868      0.0.0.0:*              LISTEN
```

Note: netstat cannot be used to check the status of Stream Control Transmission Protocol (SCTP) diameter ports.



5.10 Verify CPU Load and Memory Use

To verify the CPU load and memory use:

1. Use `ssh` to connect to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <username>@<OAM IP>
```

2. Check which nodes in the cluster that are started:

```
cdsv-get-node-state -s
```

3. Check which started nodes that have `CscfPool` or `DIASharedProcPool` hosted. If already known, continue with Step 4.

```
cdsv-print-node -v | egrep -w 'CscfPool|DIASharedProcPool' -B 3
```

Example output:

```
Result from [.cdsv.director]:
Node[0x8ebfd0] id: 9 name: safAmfNode=PL-10,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 0100100a
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedProcPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x8b9940] id: 8 name: safAmfNode=PL-11,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 0100100b
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedProcPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x876600] id: 6 name: safAmfNode=PL-12,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 0100100c
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedProcPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x822540] id: 3 name: safAmfNode=PL-3,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001003
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedProcPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x7df040] id: 1 name: safAmfNode=PL-4,
safAmfCluster=myAmfCluster state: 2 (Started)
```



```

flags: 00000001 address: 01001004
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x767b30] id: 0 name: safAmfNode=PL-5,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001005
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x8a8be0] id: 7 name: safAmfNode=PL-6,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001006
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x8658c0] id: 5 name: safAmfNode=PL-7,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001007
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x843f00] id: 4 name: safAmfNode=PL-8,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001008
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]
Node[0x7eff50] id: 2 name: safAmfNode=PL-9,
safAmfCluster=myAmfCluster state: 2 (Started)
flags: 00000001 address: 01001009
  UserApp states: [ 0: 00000003 1: 00000003 ]
  Hosted pools: [ DIASharedPool CscfPool CscfOamProcPool
CMCO_VDicosPool vDicosEEPool LpmsvCommonPool vDicosOAMPool
DicosDbClassPot_Pool DIASharedProcPool ]

```

4. Display the status of the Virtual Machines (VMs) on each node that has CscfPool or DIASharedProcPool hosted:

```
clurun.sh -c vmstatus -n <node>
```

For example:

```
clurun.sh -c vmstatus -n PL-3.lpmsv.agent.vm0
```

Example output:



Result from [PL-3.lpmsv.agent.vm0]:

Configuration:

Basic Interval: 1000 ms
Short Intervals: 1
Short Interval: 1000000 usec
Long term samples: 5

Current values:

Reconfiguration ongoing: no

Resources:

*CPU_AVG:

Core selection method: avg
Limit: 80.0%
Maint limit: 60.0%
Load: 5.0%/4.0% (<short term>/<long term>)
Shared load: 5.0%/4.0% (<short term>/<long term>)
Rate delta: -2138865795
Reject Rate: 0.000 (0)
Rejected: 0

CPU_CURRENT:

Core selection method: current
Limit: 100.0%
Maint limit: 60.0%
Load: 4.0%/2.0% (<short term>/<long term>)
Rate delta: -2130215175
Reject Rate: 0.000 (0)
Rejected: 0

CPU_MAX:

Core selection method: max
Limit: 100.0%
Maint limit: 100.0%
Load: 26.0%/32.0% (<short term>/<long term>)
Shared load: 26.0%/32.0% (<short term>/<long term>)
Rate delta: 2095940371
Reject Rate: 0.000 (0)
Rejected: 0

Memory:

Memory limit: 100%
Usage base: 69%
Memory usage: 70% (17610153984 bytes free of 57831317504 total bytes)

Scaled values:

Limit: 100.0%
Maint limit: 100.0%



```

Load: 3.0%/3.0% (<short term>/<long term>)
Shared load: 3.0%/3.0% (<short term>/<long term>)
Rate delta: -2147450880
Reject Rate: 0.000 (0)
Rejected: 0

MultiMMap:
Multimap limit: 80%
Multimap maint limit: 60%
Usage base: 4%
Multimap usage: 4% (381371 pages allocated of 8471384 total
pages)

Scaled values:
Limit: 79.0%
Maint limit: 58.0%
Load: 0.0%/0.0% (<short term>/<long term>)
Shared load: 0.0%/0.0% (<short term>/<long term>)
Rate delta: -2147450880
Reject Rate: 0.000 (0)
Rejected: 0

TIPC incoming:
Tipc overload limit: 5000
Job count: 0
Limit: 80.0%
Maint limit: 60.0%
Load: 0.0%/0.0% (<short term>/<long term>)
Rate delta: -2147450880
Reject Rate: 0.000 (0)
Rejected: 0

TIPC outgoing:
Tipc overload limit: 5000
Outgoing dialogue message count: 0
Limit: 80.0%
Maint limit: 60.0%
Load: 0.0%/0.0% (<short term>/<long term>)
Rate delta: -2147450880
Reject Rate: 0.000 (0)
Rejected: 0

Heap:
Heap limit: 80%
Heap maint limit: 60%
Usage base: 19%
Heap usage: 19% (268434432 total bytes = 52031032 used
bytes + 216403400 free bytes)

Scaled values:
Limit: 75.0%
Maint limit: 50.0%
Load: 0.0%/0.0% (<short term>/<long term>)

```



```
Rate delta:          -2147450880
Reject Rate:         0.000 (0)
Rejected:            0
```

Note: Only part of output is shown here.

5. Check the memory use, and verify that `core load` and `core load (LT)` is not more than 80%.

5.11 Verify eVIP Status

To verify the eVIP status:

1. Connect to eVIP CLI:

```
telnet `/opt/vip/bin/getactivecontrol` 25190
```

2. Display eVIP link/agent status:

```
show agents
```

3. Verify that no eVIP links/agents are `INACTIVE` and `DOWN`.

Example output:



```

+-----[ ALB alb_0 (ACTIVE) ]-----+
+-----PN-----+
| pnagent (4) | lbesel_pn (20) |
| [2] fe80::ff:fe01:e : ACTIVE | [2] fe80::ff:fe01:e : ACTIVE |
| [1] fe80::ff:fe01:b : ACTIVE | [1] fe80::ff:fe01:b : ACTIVE |
+-----+
| ersipc (0) | repdb (20) |
| [2] fe80::ff:fe01:e : ACTIVE | [2] fe80::ff:fe01:e : ACTIVE |
| [1] fe80::ff:fe01:b : ACTIVE | [1] fe80::ff:fe01:b : ACTIVE |
+-----+
| lbeagent (28) | sesel_lbe (10) |
| [2] fe80::1:f4ff:fe01:4 : ACTIVE | [2] fe80::1:f4ff:fe01:4:ACTIVE |
| [1] fe80::1:f4ff:fe01:3 : ACTIVE | [1] fe80::1:f4ff:fe01:3:ACTIVE |
+-----+
| feeagent (18) | lbesel_fe (20) |
| [2] fe80::1:f6ff:fe01:9:INACTIVE DOWN | [2] fe80::1:f6ff:fe01:9:ACTIVE |
| [1] fe80::1:f6ff:fe01:7:INACTIVE DOWN | [1] fe80::1:f6ff:fe01:7:ACTIVE |
+-----+
| sesel_fe (10) | |
| [2] fe80::1:f6ff:fe01:9 : ACTIVE | |
| [1] fe80::1:f6ff:fe01:7 : ACTIVE | |
+-----+
| seagent (18) | lbesel_se (24) |
| [2] fe80::1:f5ff:fe01:6:ACTIVE RDY | [2] fe80::1:f5ff:fe01:6:ACTIVE |
| [1] fe80::1:f5ff:fe01:5:ACTIVE RDY | [1] fe80::1:f5ff:fe01:5:ACTIVE |
+-----+
| sesel_se (6) | |
| [2] fe80::1:f5ff:fe01:6 : ACTIVE | |
| [1] fe80::1:f5ff:fe01:5 : ACTIVE | |
+-----+
| ikeagent (0) | ipsecuagent (10) |
| | [2] fe80::ff:fe01:10:ACTIVE RDY |
| | [1] fe80::ff:fe01:d:ACTIVE RDY |
+-----+
| xalbsel (4) | |
| [2] fe80::ff:fe01:f : ACTIVE | |
| [1] fe80::ff:fe01:c : ACTIVE | |
+-----+
|
|
+-----+
| eRSIP state: ACTIVE | cIPSEC state: ACTIVE RDY |
|
|
+-----+
OK

```

4. Exit eVIP CLI:

exit

5.12 Check System Environment Variables

To check the environment variables:



1. Use `ssh` to connect to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <username>@<OAM IP>
```

2. List the environment variables:

```
for envEntry in `vdicos-envdata-list | sort`; do echo \  
$envEntry = `vdicos-envdata-get $envEntry`; done;
```

Example output:

```
...  
ASSIM_SINGLEPROCESS = 1  
CSCF_DBMONITOR_MEMORY_LIMIT = 400000000  
CSCF_ENS_RESOURCE_LIMIT = 10000  
CSCF_IPSEC_DISABLED_FOR_VEGA = 1  
CX_DIAMETER_STACKID = CSCFCX  
DIA_INSTALLER_0 = CSCFCX  
DIA_INSTALLER_1 = CSCFRF  
DIA_INSTALLER_2 = CSCFRO  
DIA_INSTALLER_3 = CSCFRX  
DIA_RESOURCE_LIMIT_CSCFCX = 80000  
DIA_RESOURCE_LIMIT_CSCFRF = 80000  
DIA_RESOURCE_LIMIT_CSCFRO = 80000  
DIA_RESOURCE_LIMIT_CSCFRX = 80000  
ICMP_CONTROLLER_RESOURCE_LIMIT = 10000  
IPMM_IS_PROCESSOR_VEGA = 1  
JIMAnonPermissions = 0  
JimDebugInfo = 255  
JimMonitorsEnabled = 0  
JimTcpPortNumber = 6497  
RF_DIAMETER_STACKID = CSCFRF  
RO_DIAMETER_STACKID = CSCFRO  
RX_DIAMETER_STACKID = CSCFRX  
SIP_TIMER_T1 = 5000  
...
```

Note: Only part of output is shown here.

3. Verify that the output is as expected (with only expected deviances for system size market adaptations, and so on).

5.13 Check Availability of DNS Servers

To check the availability of the Domain Name System (DNS) servers:

1. Log on to ECLI:

```
ssh -A <username>@<OAM IP> -p <port>
```

2. Check the CSCF DNS client source IP address:



```
show ManagedElement=<nodename>,CscfFunction=1,DNS-Application=DNS,dnsLocalAddress
```

Example output:

```
dnsLocalAddress
"10.50.10.1"
```

3. Check the CSCF DNS servers:

```
show ManagedElement=<nodename>,CscfFunction=1,DNS-Application=DNS,dnsServerEntry
```

Example output:

```
dnsServerEntry
"0:137.168.10.50:53"
```

4. Log off from ECLI:

```
exit
```

5. From a controller or payload, check which nodes in the cluster that are started:

```
cdsv-get-node-state -s
```

6. Check which started nodes that have CscfPool hosted.

```
cdsv-print-node -v | egrep -w 'CscfPool' -B 3
```

Example output:

```
Result from [.cdsv.director]:
Node[0x158d1a0] id: 0 name: safAmfNode=PL-3,safAmfCluster=myAmfCluster
state: 2 (Started) flags: 00000001 address: 01001003
UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharedPool CscfPool DIASharedProcPool CscfOamProcPool
DicosDbClassPot_Pool CMCO_VDicosPool vDicosEEPool LpmsvCommonPool
vDicosOAMPool LiPool ]
Node[0x158d650] id: 1 name: safAmfNode=PL-4,safAmfCluster=myAmfCluster
state: 2 (Started) flags: 00000001 address: 01001004
UserApp states: [ 0: 00000003 1: 00000003 ]
Hosted pools: [ DIASharedPool CscfPool DIASharedProcPool CscfOamProcPool
DicosDbClassPot_Pool CMCO_VDicosPool vDicosEEPool LpmsvCommonPool
vDicosOAMPool LiPool ]
```

7. Log on to a started node that has hosted the CscfPool:

```
ssh -A <username>@<payload>
```

8. For each DNS server, run `traceroute` to check the DNS server status.



If the last hop is the destination address, the connectivity to the DNS servers is working. Otherwise, the connectivity is lost.

Note: `sudo` or `root` privileges are required for running `tracert`.

- a. If the source IP address (retrieved from the CSCF DNS configuration) is 0.0.0.0 (any):

```
tracert -I <DNS server IP address>
```

Example output:

```
tracert -I 137.168.10.50
tracert to 137.168.10.50 (137.168.10.50), 30 hops max,
60 byte packets
1  137.168.10.50 (137.168.10.50)  0.084 ms  0.013 ms  0.065 ms
```

- b. If the source IP address (retrieved from the CSCF DNS configuration) is **not** 0.0.0.0 (any):

```
tracert -I -s <source IP address> <DNS server IP address>
```

Example output:

```
tracert -I -s 10.50.10.1 137.168.10.50
tracert to 137.168.10.50 (137.168.10.50), 30 hops max,
60 byte packets
1  * * *
2  dns_src (10.50.10.1)  1.091 ms  1.078 ms  1.072 ms
3  192.168.216.6 (192.168.216.6)  1.284 ms  1.278 ms  1.268 ms
4  172.16.2.2 (172.16.2.2)  2.167 ms  2.169 ms  2.171 ms
5  172.16.2.1 (172.16.2.1)  2.431 ms  2.439 ms  2.457 ms
6  172.16.254.1 (172.16.254.1)  133.104 ms  132.076 ms  132.053 ms
7  137.168.10.50 (137.168.10.50)  1.106 ms  0.945 ms  0.919 ms
```

5.14 Check Status of SIP Interfaces

To check the status of the SIP interfaces:

1. Log on to ECLI:

```
ssh -A <username>@<OAM IP> -p <port>
```

2. Navigate to the CSCF Network Interfaces:

```
ManagedElement=<node name>,CscfFunction=1,CSCF-Application=CSCF,CscfNwIfContainer=0
```

3. For each network interface, for example `IcscfNwIfs=0` and `ScscfNwIfs=0`, check that Status is OK.:



```
show <NetworkInterface>=<transport-protocol>:<IP  
address>:<port>,<NetworkInterface>Status
```

Example:

```
show IcsfNetworkInterface=TCP:192.168.10.201:5060,ic  
scfNetworkInterfaceStatus
```

Example output:

```
icscfNetworkInterfaceStatus=OK
```

If the Status is not OK, see Section 6 on page 37.

4. Log off from ECLI:

```
exit
```

5.15 Store Health Check Report

To store the health check report: save the report in an agreed format and store it persistently.





6 Report Problems

For any abnormal situation, refer to *CSCF Troubleshooting Guideline*.

If the problem still exists, report it to the next level of support.

It is also important to collect the related data. For information about how to collect the data, refer to *Data Collection Guideline for CSCF*.





7 Example of Configuration File

Example 1 shows an example of the configuration file used by the automatic health check script.

```
# Configuration file for CscfHealthCheck
#
# Lines starting with # contain comments and are ignored.
#
# Information for logging in to cluster (controller)

# Port to be used when SSH to system controller on the cluster.
# Default port is 22
cluster.port=22

# User to be used when SSH to system controller on the cluster.
cluster.user=root

# Connection and authentication settings for ECLI.
# Address to be used when SSH to ECLI, usually the OAM VIP.
oam.host=192.168.10.200

# Port to be used when SSH to ECLI.
# Default port is 2022.
oam.ecliport=2022

# O&M user
oam.user=jambala_caa

# Settings for accessing PMF counter data.
# Counters to include by default. Repeat for multiple values.
# Format: NAME or NAME.KEY.
pmf.counters=cscfAcceptedRegistrations
pmf.counters=cscfExpiredRegistrations
pmf.counters=cscfRejectedRegistrations
pmf.counters=cscfFailedSessions
pmf.counters=cscfScscfAssignments
pmf.counters=cscfCxSelPullInitRegistrations
pmf.counters=cscfCxPullUnableToComplys
pmf.counters=cscfACABackup
pmf.counters=cscfNBASuccess
pmf.counters=cscfSipDigestAuthenticationSuccess
pmf.counters=scscfGibaSuccess

# the time at PM counter values should be collected from , could be the current time
# Time Format should be day/month/year hour:minutes [dd/mm/yy hh:mm]
start.time=None

# the time at PM counter values should be collected to , it determine the total time
# time duration for logs collection considering the start time
# Time Format should be day/month/year hour:minutes [dd/mm/yy hh:mm]
end.time=None

# Threshold that the CPU load must reach for the
# healthcheck script to flag VERIFY instead of OK.
# Default value is 81%.
cpu.max=81

#configure to select the PM log files with the specific granularity period,
#the value is in seconds
granularity.period=300
```

Example 1 Example of Configuration File





8 Example of Automatic Health Check Results

Example 2 shows an example of health check results after running automatic health check. See Section 3.2.1 Run Automatic Health Check on page 6.

```
==== CSCF Health Check
Node name: 1
Release: vCSCF_1.6.0_R7A04
Start Time: 2018-03-29 09:54:46.046193
Report(s): /storage/no-backup/vcscf_cxp9034345/healthcheck/=>
reports/CscfHealthCheckReport_1_2018-03-29_09_54_46.txt
INFO: Information for the user, not checked by the script
OK: Task passed
VERIFY: Manual verification needed
FAIL: Problem detected by the script
ERROR: An error occurred, script update needed or system broken
=====System Environment Variables=====
INFO: CSCF_DBMONITOR_MEMORY_LIMIT: 6000000000
INFO: CSCF_IPSEC_DISABLED_FOR_VEGA: 1
INFO: CX_DIAMETER_STACKID: CSCFCX
INFO: DIA_INSTALLER_0: CSCFCX
INFO: DIA_INSTALLER_1: CSCFRF
INFO: DIA_INSTALLER_2: CSCFRO
INFO: DIA_INSTALLER_3: CSCFRX
INFO: DIA_RESOURCE_LIMIT_CSCFCX: 250000
INFO: DIA_RESOURCE_LIMIT_CSCFRF: 250000
INFO: DIA_RESOURCE_LIMIT_CSCFRO: 250000
INFO: DIA_RESOURCE_LIMIT_CSCFRX: 250000
INFO: IPMM_BACKUP_PATH0: /cluster
INFO: IPMM_BACKUP_PATH1: /cluster
INFO: IPMM_BACKUP_PATH2: /cluster
INFO: IPMM_BACKUP_PATH3: /cluster
INFO: IPMM_ENABLE_BACKUP: 1
INFO: IPMM_IS_PROCESSOR_VEGA: 1
INFO: JIMAnonPermissions: 0
INFO: JimDebugInfo: 255
INFO: JimMonitorsEnabled: 0
INFO: JimTcpPortNumber: 6497
INFO: LOAD_REG_BASIC_INTERVAL: 1000
INFO: LOAD_REG_CPU_AVG_LIMIT: 80
INFO: LOAD_REG_CPU_CURRENT_LIMIT: 100
INFO: LOAD_REG_CPU_MAX_LIMIT: 100
INFO: LOAD_REG_HIST_OFF: 5
INFO: LOAD_REG_HIST_ON: 0
INFO: LOAD_REG_LIMIT: 80
INFO: LOAD_REG_LONG_TERM_SAMPLES: 5
INFO: LOAD_REG_MAINT_LIMIT: 60
INFO: LOAD_REG_MEMORY_LIMIT: 100
INFO: LOAD_REG_TIPC_OVERLOAD_LIMIT: 5000
INFO: MultiMMapMaxMem: 60
INFO: Node_Distinguished_Name: ManagedElement=jambala
INFO: RF_DIAMETER_STACKID: CSCFRF
INFO: RO_DIAMETER_STACKID: CSCFRO
INFO: RX_DIAMETER_STACKID: CSCFRX
INFO: SIP_MAX_NUM_PARSING_PROCS: 10
INFO: SIP_MSG_COUNT_LIMIT: 1000
INFO: SIP_TIMER_T1: 5000
INFO: SYSTEM_MEASUREMENT_DOMAIN: 1084266
INFO: Ss7CpManagerAddr: ss7cafcpmaddress:6669
INFO: TransactionTimerInterval: 10000
INFO: UserHeapSize: 2048
INFO: tspCmStaticTraceLevel: 0
INFO: tspCmvDicosFakeGroupId: 0
INFO: tspCmvDicosFakeUserDN: administratorName=jambala
INFO: tspCmvDicosFakeUserId: 0
INFO: vDicosLogRecordSize: 0
INFO: vDicosStaticPtAllocationSetup: Socket=*,Core=*,VT=*
```



```
INFO: vDicosVMCoreLinkSetup: Socket=*,Core=*,VT=*
VERDICT: OK
=====CSCF Network Connectivity=====
OK: CDSV Connections of the control server
OK: CDSV Connections of the distribution server
VERDICT: OK
=====EVIP=====
OK: evip status ok
VERDICT: OK
=====CSCF System State=====
OK: System State: Idle
VERDICT: OK
=====SIP Interface Status=====
OK: All SIP interfaces statuses are OK
VERDICT: OK
=====CSCF Processor Outage=====
OK: DBSv - cluster state: Idle
OK: LPMSv - cluster state: Idle
VERDICT: OK
=====PM Indicators=====
INFO: pm report is generated at /storage/no-backup/vcscf_cxp9034345/⇒
healthcheck/reports/PM_INDICATORS_Report_1_2018-03-29_09_54_46.csv
INFO: pm report is generated at /storage/no-backup/vcscf_cxp9034345/⇒
healthcheck/reports/PM_INDICATORS_Report_1_2018-03-29_09_54_46.html
VERDICT: OK
=====CSCF Operational and Administrative State=====
OK: cscfISPOperationalState=ENABLED
OK: cscfAdministrativeState=UNLOCKED
VERDICT: OK
=====CSCF Memory Usage=====
OK: PL-3: Memory Usage: 72%
OK: PL-4: Memory Usage: 72%
OK: PL-5: Memory Usage: 72%
OK: PL-6: Memory Usage: 72%
OK: PL-7: Memory Usage: 72%
OK: PL-8: Memory Usage: 72%
OK: PL-9: Memory Usage: 71%
OK: PL-10:Memory Usage: 72%
VERDICT: OK
=====CSCF configured DNS Server(s)=====
INFO: local/source: IPv4 address 10.50.43.71
OK: DNS Server 10.50.24.125 via 10.50.43.71 is reachable
VERDICT: OK
=====Diameter Port Listening=====
OK: Diameter ports are ok
VERDICT: OK
=====CSCF CPU Load=====
OK: PL-3: CPU Load Short Term: 6%
OK: Long Term: 9%
OK: PL-4: CPU Load Short Term: 7%
OK: Long Term: 9%
OK: PL-5: CPU Load Short Term: 7%
OK: Long Term: 8%
OK: PL-6: CPU Load Short Term: 9%
OK: Long Term: 11%
OK: PL-7: CPU Load Short Term: 5%
OK: Long Term: 7%
OK: PL-8: CPU Load Short Term: 10%
OK: Long Term: 10%
OK: PL-9: CPU Load Short Term: 6%
OK: Long Term: 8%
OK: PL-10:CPU Load Short Term: 9%
OK: Long Term: 11%
VERDICT: OK
=====Controller status - SC-1=====
OK: ro:Secondary/Primary -- This SC is Secondary
OK: cs:Connected
OK: ds:UpToDate/UpToDate
OK: This controller is STANDBY on CoreMW level
VERDICT: OK
=====Controller status - SC-2=====
OK: ro:Primary/Secondary -- This SC is Primary
OK: cs:Connected
OK: ds:UpToDate/UpToDate
OK: This controller is ACTIVE on CoreMW level
VERDICT: OK
```



```
=====FM Alarms and Notifications=====
VERIFY: MINOR: ManagedElement=1,Stack=CSCFRX, "vDicos, Diameter =>
Own Node Disabled"
FAIL: MAJOR: ManagedElement=1,SystemFunctions=1,Lm=1, "License =>
Management, Autonomous Mode Activated"
FAIL: MAJOR: ManagedElement=1,Conn=conn1,Stack=CSCFRO,=>
Host=LABSPTONCHA.ericsson.se, "vDicos, Diameter Link Failure"
VERDICT: FAIL
=====Total Verdict=====
VERDICT: FAIL
```

Example 2 Example of Automatic Health Check Result





9 File Management

The Health Check report files are exposed by File Management in the following file group structure:

- FileGroup=Cscf
 - FileGroup=HealthCheck
 - FileGroup=ReportFiles

For more information on file groups, refer to *Handling Files*.