

# CSCF Health Check

Call Session Control Function

OPERATING INSTRUCTIONS

**Copyright**

© Ericsson AB 2016–2019. 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>Introduction</b>	<b>1</b>
1.1	Prerequisites	1
<b>2</b>	<b>Health Check Tasks</b>	<b>3</b>
<b>3</b>	<b>Automated Health Check Procedures</b>	<b>5</b>
3.1	Preparations	5
3.2	Automatic Health Check	5
3.3	Automatic Health Check Verdict	6
3.4	Automatic Health Check Options	7
<b>4</b>	<b>Configuration File</b>	<b>11</b>
4.1	Cluster Port	11
4.2	Granularity Period	11
4.3	O&M Port	11
4.4	Maximum CPU Load	11
4.5	Counters	12
<b>5</b>	<b>Manual Health Check Procedure</b>	<b>13</b>
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	24
5.11	Verify eVIP Status	26
5.12	Check System Environment Variables	28
5.13	Check Availability of DNS Servers	30
5.14	Check Status of SIP Interfaces	31
5.15	Store Health Check Report	32



<b>6</b>	<b>Report Problems</b>	<b>33</b>
<b>7</b>	<b>Example of Configuration File</b>	<b>35</b>
<b>8</b>	<b>Example of Automatic Health Check Results</b>	<b>37</b>
<b>9</b>	<b>Examples of Scheduling Automatic Health Checks</b>	<b>39</b>
<b>10</b>	<b>File Management</b>	<b>41</b>



# 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), see [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:

- Knowing the CSCF system on a System administrator level.
- Knowing the OAM Movable IP address.
- Knowing the logon credentials of an Operation & Maintenance (O&M) user that has sufficient rights to display CSCF Configuration Management (CM) and Fault Management (FM) parameters. For example, an O&M user of the System Troubleshooter role.





## 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:

1. `ssh <username>@<OAM-MIP>`

#### 3.1.2 Obtain Persistent Storage Area Paths

To obtain the different Persistent Storage area paths for the system:

1. 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 an 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 11.

2. Check the results printed to the console. An example of the results is shown in Section 8 on page 37.
3. Find the generated report files with the most recent date and time in the directory:

```
<storage-path>/vcscf_cxp9034345/healthcheck/reports/<vnftype>_H
C_<vnfname>_<timestamp>_<type>.html
```

```
<storage-path>/vcscf_cxp9034345/healthcheck/reports/<vnftype>_H
C_<vnfname>_<timestamp>_<type>.xml
```

```
<storage-path>/vcscf_cxp9034345/healthcheck/reports/PM_INDICAT
ORS_Report_<nodename>_<time stamp>.html
```

Health Check report files can be fetched using File Management. For more information, see Section 10 on page 41.

**Note:** The time stamp of the generated healthcheck report files has format <YYYY-MM-DDTHH:MM:SSshh:mm>.

The time stamp of the generated PM\_INDICATORS report file has format <YYYY-MM-DD\_HH\_MM\_SS>.

## 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. The definitions of the final Health Check verdicts are shown in Table 3.



Table 2 Verdict Definitions for Individual Checks

Verdict	Description
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.

Table 3 Final Verdict Definitions

Verdict	Description
HEALTHY	All checks are successful. All individual checks have an INFO or OK verdict.
NOT_HEALTHY	At least one individual check has a VERIFY, FAIL, or ERROR verdict in the health check procedure.

## 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 4.



Table 4 Command Line Options Supported by Health Check Script

Option	Meaning
<b>-check CHECK [CHECK]</b>	<p>Specifies the checks to run.</p> <p>These are possible values:</p> <ul style="list-style-type: none"><li>• <b>cscfprocessoroutage</b></li><li>• <b>pmindicator</b></li><li>• <b>sipinterfaces</b></li><li>• <b>cscfnetworkconnectivity</b></li><li>• <b>cscfmemusage</b></li><li>• <b>cscfcpuload</b></li><li>• <b>controllerstatus</b></li><li>• <b>systemenvironmentvariables</b></li><li>• <b>cscfdnsservers</b></li><li>• <b>alarms</b></li><li>• <b>cscfisopstateandadminstate</b></li><li>• <b>cscfsystemstate</b></li><li>• <b>evipstatus</b></li><li>• <b>diameterports</b></li></ul> <p>It is possible to give multiple checks as input, separated with whitespaces, commas, or a combination of both.</p>
<b>--cpu-max=CPU_MAX</b>	Sets the threshold, in %, that the CPU load must reach for the healthcheck script to flag VERIFY instead of OK.
<b>-h, --help</b>	Prints a brief help message and exits.
<b>-q, quiet</b>	Prints only the verdict for each check in console.
<b>-r REPORT, --report REPORT</b>	Specifies the location to save report. The default is <storage-path>/vcscf_cxp9034345/healthcheck/reports
<b>-sd, -schedule_delete</b>	<p>Deletes one or more periodic health checks.</p> <p>The value is the identity of the health check. If more periodic health checks are to be deleted, separate the identities by comma: &lt;id1, id2, ...&gt;. If all periodic health checks are to be deleted, specify the value as <b>all</b>.</p>
<b>-sg, -schedule_get</b>	Shows active periodic health checks and the respective identities for the CSCF.



Table 4 Command Line Options Supported by Health Check Script

Option	Meaning
<b>-sp, -schedule_period</b>	<p>Specifies an interval for running a health check periodically. The health check is performed with the defined interval until it is deleted with option <b>-schedule_delete</b>.</p> <p>The values are 1, 2, 3, 4, 6, 8, 12, 24 hours.</p>
<b>-ss, -schedule_start</b>	<p>Defines when the periodic health check starts. The periodic health check starts at the next occurrence of the given time.</p> <p>The value is specified in <b>HH:MM</b>. For example, <b>11:12</b>. The default value is the time of the issued command.</p>
<b>-type USECASE</b>	<p>Specifies the use case to run.</p> <p>The possible values are <b>basic</b> and <b>full</b>.</p>
<b>-verbose</b>	Prints the used input commands and their raw output.





## 4 Configuration File

The automatic health check script uses a configuration file that contains parameters that can be configured by the user.

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.

The configuration file is stored in `<config-path>/vcscf_cxp9034345/health check/`, and is called `<user-name>.config`. The file permission is set to read and write for the user.

If the configuration file does not exist, the automatic health check script creates a configuration file with default values.

### 4.1 Cluster Port

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

This parameter is optional. The default value is 1022.

### 4.2 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 is 300 seconds.

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

### 4.3 O&M Port

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

This parameter is optional. The default value is 2022.

### 4.4 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%.



## 4.5 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:

1. **ssh <username>@<OAM-MIP>**

### 5.2 Check Release Information

For more information regarding the ECLI, see [Ericsson Command-Line Interface User Guide](#).

To check the CSCF Release Information:

1. Log on to ECLI:

```
ssh <username>@<OAM-MIP> -p <port>
```

2. Check the release parameter in ManagedElement:

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

Example output:

```
ManagedElement=1
managedElementType="vCSCF "
release="1.11.0"
CscfFunction=1
[...]
```

The following example output shows an vCSCF 1.10 Emergency Package (EP) release.

```
ManagedElement=1
managedElementType="vCSCF "
release="1.10.1"
CscfFunction=1
Equipment=1
SystemFunction
```



**Note:** The example shows the vCSCF 1.10 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="CXP9034345/1 R12A (1.11.0-8)"
  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
```

4. Log off from the ECLI:

```
exit
```

## 5.3 Verify Status of Alarms

For more information regarding the ECLI, see [Ericsson Command-Line Interface User Guide](#).

To verify the status of alarms:

1. Log on to ECLI:

```
ssh <username>@<OAM-MIP> -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 Instruction.

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:

```
immlist lpmsv=LPMSvSite
```

For more information, see [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 <emergency_username>@<OAM-MIP>
```

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 <emergency_username>@<OAM-MIP>
```

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, see [Ericsson Command-Line Interface User Guide](#).

To verify that the CSCF is ready to handle traffic:

1. Log on to ECLI:

```
ssh <username>@<OAM-MIP> -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 <emergency_username>@<OAM-MIP>
```

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

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

3. Log on to ECLI:

```
ssh <username>@<OAM-MIP> -p <port>
```

4. Check `portNr`, `ipAddressesList`, and `sctpAddressesList` for each configured diameter interface. If already known, continue with Step 7.

```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-Applica
tion=DIA,DIA-CFG-StackContainer=CSCFCX,DIA-CFG-OwnNodeConfig=
CSCFCX
```

```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-Applica
tion=DIA,DIA-CFG-StackContainer=CSCFRF,DIA-CFG-OwnNodeConfig=
CSCFRF
```



```
show ManagedElement=<nodename>,CscfFunction=1,DIA-CFG-Applica
tion=DIA,DIA-CFG-StackContainer=CSCFR0,DIA-CFG-OwnNodeConfig=
CSCFR0
```

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"
```

5. Make a notation of portNr, ipAddressesList, and sctpAddressesList.
6. Log off from ECLI:

```
exit
```

7. 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. Log on to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <emergency_username>@<OAM-MIP>
```

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

```
cdsd-get-node-state -s
```

3. Display the status of the Virtual Machines (VMs) on each node:

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

For example:

```
clurun.sh -c vmstatus -n PL-3
```

Example output:

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

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 the output is shown here.

4. Check that the following items are not more than 80%:
  - In Memory: memory usage
  - In MultiMMap: multimap use, the allocated memory
  - In \*CPU\_AVG, CPU\_CURRENT, and CPU\_MAX: load for short term and long term
5. Log off from the PL:
 

```
exit
```

## 5.11 Verify eVIP Status

To verify the eVIP status:

1. Log on to the 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 -----+
| pagent (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 |
+-----+
+----- LBE -----+
| 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 |
+-----+
+----- FE -----+
| 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 | |
+-----+
+----- SE -----+
| 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 | |
+-----+
+----- IPSEC -----+
| ikeagent (0) | ipsecuagent (10) |
| | [2] fe80::ff:fe01:10:ACTIVE RDY |
| | [1] fe80::ff:fe01:d:ACTIVE RDY |
+-----+
+----- XALBSEL -----+
| xalbsel (4) | |
|[2] fe80::ff:fe01:f : ACTIVE | |
|[1] fe80::ff:fe01:c : ACTIVE | |
+-----+
+-----+
eRSIP state: ACTIVE cIPSEC state: ACTIVE RDY
OK
```

4. Log off from the eVIP CLI:

**exit**



## 5.12 Check System Environment Variables

To check the environment variables:

1. Log on to a Payload (for example PL-3) using the cluster user and password:

```
ssh -A <emergency_username>@<OAM-MIP>
```

2. List the environment variables:

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

Example output:





```

CSCF_DBMONITOR_MEMORY_LIMIT = 600000000
CSCF_IPSEC_DISABLED_FOR_VEGA = 1
CX_DIAMETER_STACKID = CSCFCX
DIA_INSTALLER_0 = CSCFCX
DIA_INSTALLER_1 = CSCFRF
DIA_INSTALLER_2 = CSCFRO
DIA_RESOURCE_LIMIT_CSCFCX = 250000
DIA_RESOURCE_LIMIT_CSCFRF = 250000
DIA_RESOURCE_LIMIT_CSCFRO = 250000
IPMM_BACKUP_PATH0 = /storage/no-backup
IPMM_BACKUP_PATH1 = /storage/no-backup
IPMM_BACKUP_PATH2 = /storage/no-backup
IPMM_BACKUP_PATH3 = /storage/no-backup
IPMM_ENABLE_BACKUP = 1
IPMM_IS_PROCESSOR_VEGA = 1
JIMAnonPermissions = 0
JimDebugInfo = 255
JimMonitorsEnabled = 0
JimTcpPortNumber = 6497
LI_STANDARD = 1
LOAD_REG_BASIC_INTERVAL = 1000
LOAD_REG_CPU_AVG_LIMIT = 80
LOAD_REG_CPU_CURRENT_LIMIT = 100
LOAD_REG_CPU_MAX_LIMIT = 100
LOAD_REG_HIST_OFF = 5
LOAD_REG_HIST_ON = 0
LOAD_REG_LIMIT = 80
LOAD_REG_LONG_TERM_SAMPLES = 5
LOAD_REG_MAINT_LIMIT = 70
LOAD_REG_MEMORY_LIMIT = 100
LOAD_REG_MIN_TARGETCALLS = 100
LOAD_REG_TIPC_OVERLOAD_LIMIT = 5000
MultiMMapMaxMem = 60
Node_Distinguished_Name = ManagedElement=jambala
PM_COLLECTOR_FLUSH_PERIOD = 5
RF_DIAMETER_STACKID = CSCFRF
RO_DIAMETER_STACKID = CSCFRO
SIP_MAX_NUM_PARSING_PROCS = 10
SIP_MSG_COUNT_LIMIT = 1000
SIP_TIMER_T1 = 5000
Ss7CpManagerAddr = ss7cafcpmaddress:6669
SYSTEM_MEASUREMENT_DOMAIN = 1084266
TransactionTimerInterval = 10000
tspCmStaticTraceLevel = 0
tspCmvDicosFakeGroupId = 0
tspCmvDicosFakeUserDN = administratorName=jambala
tspCmvDicosFakeUserId = 0
UserHeapSize = 2048
vDicosLogRecordSize = 0
vDicosVMCoreLinkSetup = Socket=*,Core=*,VT=*

```



**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).
4. Log off from the PL:

**exit**

## 5.13 Check Availability of DNS Servers

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

1. Log on to ECLI:

```
ssh <username>@<OAM-MIP> -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. Log on to a PL from one of the SCs:

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

7. For each DNS server, run the dig command to check the DNS server status.

- a. Run the dig command to test a DNS server availability:



```
dig @<DNS server IP address> -p <DNS server port> a
host.availability.test -b <DNS Local address>
```

Example output:

```
; <<>> DiG 9.9.9-P1 <<>> @192.168.12.50 -p 53 a =>
host.availability.test -b 192.168.12.1
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NXDOMAIN, id: 35306
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;host.availability.test.      IN      A

;; AUTHORITY SECTION:
availability.test. 1800 IN SOA ns.availability.test.
\root.localhost.availability.test. 2 28800 7200 604800 86400

;; Query time: 20 msec
;; SERVER: 192.168.12.50#5353(192.168.12.50)
;; WHEN: Tue Dec 11 16:11:12 CET 2018
;; MSG SIZE rcvd: 105
```

- b. Check the return code of the last process of the dig command:

```
echo $?
```

- c. Is the return code 0?

Yes: The DNS server is up.

No: The DNS server is down.

8. Log off from the node:

```
exit
```

## 5.14 Check Status of SIP Interfaces

To check the status of the SIP interfaces:

1. Log on to ECLI:

```
ssh <username>@<OAM-MIP> -p <port>
```

2. Navigate to the CSCF Network Interfaces:



```
ManagedElement=<node name>,CscfFunction=1,CSCF-Application=CS  
CF,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 IcscfNetworkInterface=TCP:192.168.10.201:5060,icscfNetwo  
rkInterfaceStatus
```

Example output:

```
icscfNetworkInterfaceStatus=OK
```

If the Status is not OK, see Section 6 on page 33.

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, see [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, see [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

# 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

# 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

```

==== CSCF Health Check
Node name: 1
Release: CXP9034345/1 R12A08 (1.11.0-8)
Start Time: 2019-06-18 13:58:11.092630
Report(s): /storage/no-backup/vcscf_CXP9034345/healthcheck/reports/\
CscfHealthCheckReport_1_2019-06-18_13_58_11.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: 600000000
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: CSCFR0
INFO: DIA_RESOURCE_LIMIT_CSCFCX: 250000
INFO: DIA_RESOURCE_LIMIT_CSCFRF: 250000
INFO: DIA_RESOURCE_LIMIT_CSCFR0: 250000
INFO: IPMM_BACKUP_PATH0: /storage/no-backup
INFO: IPMM_BACKUP_PATH1: /storage/no-backup
INFO: IPMM_BACKUP_PATH2: /storage/no-backup
INFO: IPMM_BACKUP_PATH3: /storage/no-backup
INFO: IPMM_ENABLE_BACKUP: 1
INFO: IPMM_IS_PROCESSOR_VEGA: 1
INFO: JIMAnonPermissions: 0
INFO: JimDebugInfo: 255
INFO: JimMonitorsEnabled: 0
INFO: JimTcpPortNumber: 6497
INFO: LI_STANDARD: 1
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: 70
INFO: LOAD_REG_MEMORY_LIMIT: 100
INFO: LOAD_REG_MIN_TARGETCALLS: 100
INFO: LOAD_REG_TIPC_OVERLOAD_LIMIT: 5000
INFO: MultiMMapMaxMem: 60
INFO: Node_Distinguished_Name: ManagedElement=jambala
INFO: PM_COLLECTOR_FLUSH_PERIOD: 5
INFO: RF_DIAMETER_STACKID: CSCFRF
INFO: RO_DIAMETER_STACKID: CSCFR0
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: 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_2019-06-18_13_58_11.csv
INFO: pm report is generated at /storage/no-backup/vcscf_CXP9034345/healthcheck/reports/\
PM_INDICATORS_Report_1_2019-06-18_13_58_11.html
VERDICT: OK
=====CSCF Operational and Administrative State=====
OK: cscfISPOperationalState=ENABLED
OK: cscfAdministrativeState=UNLOCKED
VERDICT: OK
=====CSCF Memory Allocation And Usage=====
OK: PL-3:Memory Usage: 18%
OK: PL-3:Memory Allocation: 73%
OK: PL-4:Memory Usage: 18%
OK: PL-4:Memory Allocation: 73%
OK: PL-5:Memory Usage: 18%
OK: PL-5:Memory Allocation: 73%
OK: PL-6:Memory Usage: 20%
OK: PL-6:Memory Allocation: 73%
OK: PL-7:Memory Usage: 20%
OK: PL-7:Memory Allocation: 73%
OK: PL-8:Memory Usage: 20%
OK: PL-8:Memory Allocation: 73%
VERDICT: OK
=====CSCF configured DNS Server(s)=====
INFO: local/source: IPv4 address 10.50.54.49
OK: 3.2.0.219 port 53 via 10.50.54.49 | DNS-LOOKUP: Succeeded
VERDICT: OK
=====Diameter Port Listening=====
OK: Diameter ports are ok
VERDICT: OK
=====CSCF CPU Load=====
OK: PL-3:CPU Load Short Term: 34%
OK: Long Term: 38%
OK: PL-4:CPU Load Short Term: 33%
OK: Long Term: 32%
OK: PL-5:CPU Load Short Term: 30%
OK: Long Term: 29%
OK: PL-6:CPU Load Short Term: 31%
OK: Long Term: 30%
OK: PL-7:CPU Load Short Term: 34%
OK: Long Term: 37%
OK: PL-8:CPU Load Short Term: 29%
OK: Long Term: 31%
VERDICT: OK
=====Controller status - SC-1=====
OK: ro:Primary -- This SC is Primary
OK: cs:Connected
OK: ds:UpToDate/UpToDate
OK: This controller is ACTIVE on CoreMW level
VERDICT: OK
=====Controller status - SC-2=====
OK: ro:Secondary -- This SC is Secondary
OK: cs:Connected
OK: ds:UpToDate/UpToDate
OK: This controller is STANDBY on CoreMW level
VERDICT: OK
=====FM Alarms and Notifications=====
OK: No Alarms found
VERDICT: OK
=====Total Verdict=====
VERDICT: OK
```

## Example 2 Example of Automatic Health Check Result



## 9 Examples of Scheduling Automatic Health Checks

This section gives the following examples of scheduling automatic health checks:

- Example 3 shows how to schedule a health check with a defined start time.
- Example 4 shows how to schedule a periodic health check with a defined interval that starts immediately.
- Example 5 shows how to schedule a periodic health check with a defined interval and a defined start time.
- Example 6 shows how to delete scheduled health checks using the identities, such as 1.
- Example 7 shows how to delete scheduled health checks using the option all.

```
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_start 10:10
Enter current user password:
The following HealthCheck run is scheduled:
1:schedule_start=2018-12-05T10:10 type=basic time_stamp_of_command=2018-12-04T16:15
```

### Example 3 Scheduling a Health Check with a Start Time

```
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_period 8
Enter current user password:
The following HealthCheck run is scheduled:
2:schedule_start=2018-12-04T16:17 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:15
```

### Example 4 Scheduling a Periodic Health Check

```
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_start 20:20 -schedule_period 8
Enter current user password:
The following HealthCheck run is scheduled:
3:schedule_start=2018-12-04T20:20 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:16
```

### Example 5 Scheduling a Periodic Health Check with a Start Time

```
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_get
The list of scheduled HealthCheck runs:
1:schedule_start=2018-12-05T10:10 type=basic time_stamp_of_command=2018-12-04T16:15
2:schedule_start=2018-12-04T16:17 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:15
3:schedule_start=2018-12-04T20:20 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:16
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_delete 1,3,5
No schedule with the following IDs:
5
Enter current user password:
The following schedules are deleted
3:schedule_start=2018-12-04T20:20 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:16
1:schedule_start=2018-12-05T10:10 type=basic time_stamp_of_command=2018-12-04T16:15
```

### Example 6 Deleting Scheduled Health Checks Using the Identities



```
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_get
The list of scheduled HealthCheck runs:
2:schedule_start=2018-12-04T16:17 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:15
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_delete all
Enter current user password:
The following schedules are deleted
2:schedule_start=2018-12-04T16:17 schedule_period=8 type=basic time_stamp_of_command=2018-12-04T16:15
tester@SC-1:/<emergency_username>> cscfHealthCheck -schedule_get
No HealthCheck is scheduled
```

### Example 7 Deleting Scheduled Health Checks Using Option All



## 10 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, see [Handling Files](#).