

CUDB Health Check

USER GUIDE

Copyright

© Ericsson AB 2016. 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	Revision Information	1
1.2	Prerequisites	1
1.3	Related Information	2
2	Health Check Tasks	3
3	Health Check Procedure	5
3.1	Status of the CUDB System	5
3.2	UDC Cockpit Tool for Health Check	10
3.3	Detailed Health Check Procedures	11
4	Problem Reporting	15
	Glossary	17
	Reference List	19





1 Introduction

This document describes how to perform the health check procedure on the CUDB node.

A health check is performed to verify that no degradations have been introduced into the network after procedures such as reconfiguration, software updates and software upgrades. A health check can also be performed during emergencies to quantify the problems in the network. When changes are made in the network, data used for verification must be collected manually, both before and after the change.

The health check procedures are recommended to be performed before and after a system update/upgrade, a normal backup, or during periodic maintenance. They can also be run as basic normality checks.

1.1 Revision Information

Rev. A

This document is based on 7/1543-HDA 104 03/9 with the following changes:

- Removed obsolete information throughout the document.
- Virtualization terminology updates throughout the document.
- Section 3.1.1 on page 5 and Section 3.1.1.1 on page 10: Added OAM1 to LDAP counter and [-W-] LDAP counter in Example 8 and Example 11, respectively.
- Section 3.3.2 on page 11: Updated OAM IP addresses.

1.2 Prerequisites

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

1.2.1 Documents

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

- This document.



- All the documents listed in the References section, see *References*, Reference List on page 19.

1.2.2 Tools

The following tool can be used for the health check:

- UDC Cockpit

1.2.3 Conditions

Before health check can be performed, the following conditions must be met.

The users who perform the Health Check must possess the followings:

- Basic knowledge of the CUDB System.
- Knowledge of the IP addresses for the CUDB nodes.
- Required passwords. Refer to *CUDB Users and Passwords*, Reference [1] for more information on the required users and passwords.

1.3 Related Information

Definition and explanation of acronyms and terminology, trademark information, and typographic conventions can be found in the following documents:

- *CUDB Glossary of Terms and Acronyms*, Reference [2]
- *Trademark Information*, Reference [3]
- *Typographic Conventions*, Reference [4]



2 Health Check Tasks

To determine the node health, the following must be verified or checked:

- The CUDB software version.
- The Blackboard Coordination (BC) cluster status.
- The System Monitor (SM) status.
- The status of the master and slave clusters.
- The status of the Data Store (DS).
- The status of replication and the active channel.
- The raised alarms.
- The connection to the database cluster servers.
- The running CUDB processes.

These tasks can be checked by executing the `cudbSystemStatus` command. See Section 3.1 on page 5 for more information about the command.





3 Health Check Procedure

This section describes the procedures for determining the health of the node.

3.1 Status of the CUDB System

The `cudbSystemStatus` command is the primary tool for the Health Check. Use this command to get status information on the CUDB system.

To execute a normality health check on the CUDB system, run the following command:

```
cudbSystemStatus
```

For use and help regarding the command, run the following command:

```
cudbSystemStatus -h
```

3.1.1 Output of `cudbSystemStatus`

This section provides an example output of the `cudbSystemStatus` command. The output is divided into several blocks, each having an explanation below the fault examples. Refer to *CUDB Node Commands and Parameters*, Reference [5] for more information about the command.

Example 1 shows the first part of the command output, listing the available System Monitor (SM) leaders:



```
CUDB10 SC_2_1# cudbSystemStatus

Execution date: Thu Aug 30 13:52:48 CEST 2012

CUDB Software Version:
!- CUDB DESIGN DISTRIBUTION: CUDB13B CXP9020214/6 R1B549

Checking BC clusters:

[Site 1]

SM leader: Node 10 OAM1

Node 172.80.123.10
  BC server in SC_2_1 ..... running
  BC server in SC_2_2 ..... running (Leader)
  BC server in PL_2_5 ..... running

[Site 2]

SM leader: Node 11 OAM1

Node 172.80.123.11
  BC server in SC_2_1 ..... running
  BC server in SC_2_2 ..... running (Leader)
  BC server in PL_2_5 ..... running

[Site 3]

SM leader: Node 9 OAM2

Node 172.80.123.9
  BC server in SC_2_1 ..... running
  BC server in SC_2_2 ..... running (Leader)
  BC server in PL_2_5 ..... running

Checking System Monitor BC status in local node:

      SM-BC in OAM1 ..... running
      SM-BC in OAM2 ..... running

Example 1 cudbSystemStatus Output Listing the SM Leaders
```

Do!

If no SM leader is running on any of the sites, contact the next level of support **immediately**.



Note: In case the CUDB system is deployed on native BSP 8100 hardware with different hardware types, the output will show the used hardware types immediately after the CUDB software version block, as shown in Example 2.

```
CUDB10 SC_2_1# cudbSystemStatus

Execution date: Sat Mar 12 14:43:28 CET 2016

CUDB Software Version:
!- CUDB DESIGN DISTRIBUTION: CUDB13B CXP9020214/6 R1B549

Checking Hardware Type:
This system is working on following hardware types: EBS_GEP3, EBS_GEP5.
...
```

Example 2 cudbSystemStatus Output Listing the Hardware Types

Example 3 shows the second part of the command output, listing the status of the Processing Layer Database (PLDB) and Data Store Unit Group (DSG) clusters:

```
Checking Clusters status:
Node 9:
  PL Cluster (29%) .....OK
  DSG1 Cluster (23%) .....OK
  DSG255 Cluster (23%) .....OK
Node 10:
  PL Cluster (29%) .....OK
  DSG1 Cluster (23%) .....OK
  DSG255 Cluster (23%) .....OK
Node 11:
  PL Cluster (29%) .....OK
  DSG1 Cluster (22%) .....OK
  DSG255 Cluster (22%) .....OK

Checking NDB status:
  PL NDB's (2/2) .....OK
  DS1 NDB's (2/2) .....OK
  DS2 NDB's (2/2) .....OK
```

Example 3 cudbSystemStatus Output Listing the Cluster Status

If any of the clusters shows a different value than **OK**, check the active alarms and refer to the related Alarm Operating Instructions (OPIs).

Example 4 shows the third part of the command output, listing the status of the replication channels in the system.



Checking Replication Channels in the System:

Node	9	10	11
PLDB	S1	S1	M
DSG 1	S1	S1	M
DSG 255	S1	S1	M

Printing Detailed Replication Status for the Slave Replicas:

```
Node 9:
  Replication in DSG0(Chan=1)    .... Up -- Delay = 0.0 seconds,    no. of pending changes = 0
  Replication in DSG1(Chan=1)    .... Up -- Delay = 0.0 seconds,    no. of pending changes = 0
  Replication in DSG255(Chan=1)  .... Up -- Delay = 0.0 seconds,    no. of pending changes = 0
Node 10:
  Replication in DSG0(Chan=1)    .... Up -- Delay = 0.0 seconds,    no. of pending changes = 0
  Replication in DSG1(Chan=1)    .... Up -- Delay = 0.0 seconds,    no. of pending changes = 0
  Replication in DSG255(Chan=1)  .... Up -- Delay = 0.0 seconds,    no. of pending changes = 0
Node 11:
  There are no Slave clusters
```

Example 4 cudbSystemStatus Output Listing the Replication Channels

See Example 9 for an example of replication channel faults.

Example 5 shows the fourth part of the output, listing the active alarms raised by the system:

```
Printing Alarms...
[Aug 30 12:50:05] ( Preventive Maintenance Logchecker has \
found major error(s). )
```

Example 5 cudbSystemStatus Output Listing the Active Alarms

If the Printing Alarms segment shows any alarms, check their related Alarm OPIs.

Example 6 shows the fifth part of the output, listing the status of the database cluster server connections:

```
Checking MySQL server connection:
MySQL Master Servers connection .....OK
MySQL Slave Servers connection .....OK
MySQL Access Servers connection .....OK
```

Example 6 cudbSystemStatus Output Listing Database Cluster Server Connections

If any of the database cluster connections shows a value other than **OK**, check the active alarms and follow the related Alarm OPIs. See Example 10 for an example of database cluster server connection faults.

Example 7 shows the sixth part of the command output, listing the status of the Cluster Supervisor and the Blackboard Coordination (BC) SM:



Checking Process:

OAMs.....

```
Cluster Supervisor.....Running
System Monitor BC.....Running
```

Example 7 *cudbSystemStatus* Output Listing the Cluster Supervisor and the BC SM

Do!

If any of the above processes is indicated as *Not Running*, contact the next level of Ericsson support **immediately**.

Example 8 shows the final part of the command output, listing the status of the running processes:

```
OAMs.....
Cluster Supervisor.....Running
System Monitor BC.....Running
Reconciliation process.....Running in: OAM2
Smp-client.....Running
Management Server Process (ndb_mgmd).....Running
KeepAlive process.....Running
ESA.....Running
LDAP counter.....Running in: OAM1
Log Handler process.....Running
PLs.....
Storage Engine process (ndbd).....Running
LDAP FE.....Running
KeepAlive process.....Running
MySQL server process (Master).....Running
MySQL server process (Slave).....Running
MySQL server process (Access).....Running
CudbNotifications process.....Running
LDAP FE Monitor process.....Running
DSs.....
Storage Engine process (ndbd).....Running
LDAP FE.....Running
KeepAlive process.....Running
MySQL server process (Master).....Running
MySQL server process (Slave).....Running
MySQL server process (Access).....Running
LDAP FE Monitor process.....Running
```

Example 8 *cudbSystemStatus* Output Listing the Running Processes

Do!

If any of the above processes are shown as *Not Running*, contact the next level of Ericsson support.



See Example 11 for an example of process faults.

3.1.1.1 Fault Examples

This section shows examples of the faults that can be indicated in the output of the `cudbSystemStatus` command. If any of these faults occurs, contact the next level of support. The examples are as follows:

```

=====|Node 45 |Node 46
=====|=====
[-E-] PLDB _____|Xm____|Xu____
[-E-] DSG 1 _____|Xm____|Xu____
[-E-] DSG 2 _____|Xm____|Xu____

Printing Detailed Replication Status for the Slave Replicas:
Node 45:
    There are no Slave clusters
Node 46:
    There are no Slave clusters

```

Example 9 Replication Channels Down

```
[-W-] MySQL Slave Server connection Fault in.....: PL_2_3
```

Example 10 Database Cluster Server Connection Fault

```

OAMs.....
[-W-] Cluster Supervisor.....Not running in: OAM2
[-W-] System Monitor BC.....Not running in: OAM2
[-W-] Reconciliation process.....Not running in: OAM1      OAM2
[-W-] Smp-client.....Not running in: OAM2
[-W-] Management Server Process (ndb_mgmd).....Not running in: OAM2
[-W-] KeepAlive process.....Not running in: OAM2
[-W-] ESA.....Not running in: OAM2
[-W-] LDAP counter.....Not running in: OAM1      OAM2
[-W-] Log Handler process.....Not running in: OAM2
PLs.....
[-W-] Storage Engine process (ndbd).....Not running in: PL0
[-W-] LDAP FE.....Not running in: PL0
[-W-] KeepAlive process.....Not running in: PL0
[-W-] MySQL server process (Master).....Not running in: PL0
[-W-] MySQL server process (Slave).....Not running in: PL0
[-W-] MySQL server process (Access).....Not running in: PL0
[-W-] CudbNotifications process.....Not running in: PL0
[-W-] LDAP FE Monitor process.....Not running in: PL0
DSs.....
[-W-] Storage Engine process (ndbd).....Not running in: DS2_1
[-W-] LDAP FE.....Not running in: DS2_1
[-W-] KeepAlive process.....Not running in: DS2_1
[-W-] MySQL server process (Master).....Not running in: DS2_1
[-W-] MySQL server process (Slave).....Not running in: DS2_1
[-W-] MySQL server process (Access).....Not running in: DS2_1
[-W-] LDAP FE Monitor process.....Not running in: DS2_1

```

Example 11 Processes Not Running in the Node

3.2 UDC Cockpit Tool for Health Check

To follow present and recall earlier system status and performance information of CUDB nodes, use the UDC Cockpit tool. This is a monitoring application, which presents collected data on a single, web-based GUI.



3.3 Detailed Health Check Procedures

This section describes the detailed procedures for determining the health of the node.

3.3.1 Checking Active Alarms

The alarm list can be checked through the Operational Support System (OSS) or by executing the `fmactivealarms` command as shown in the example below:

```
SC_2_1# fmactivealarms
```

3.3.2 Checking ESA Processes

Use the `esa status` command to check if the ESA processes are running in both SCs. All ESA agents must be running. See the example below on how to run the command and what the expected output looks like:

```
SC_2_1# esa status
```

Expected output:

```
[info] ESA Sub Agent is running.  
[info] ESA Master Agent is running.  
[info] ESA PM Agent is running.
```

```
SC_2_1# ssh OAM2 esa status
```

Expected output:

```
[info] ESA Sub Agent is running.  
[info] ESA Master Agent is running.  
[info] ESA PM Agent is running.
```

Check the ESA cluster status with the following command on any of the SCs:

```
esaclusterstatus
```

The expected output must look similar to the example below. One SC must be in `M` state, while the other in `(M)` state:

```
M * OAM1 10.22.0.1  
(M) OAM2 10.22.0.2
```

The description of the different states are:

M ESA Master is located in that SC.



(M) ESA Slave is located in that SC.

* The SC from where the command was sent.

3.3.3 Checking Database Cluster Load

The drop ratio of the PLDB and the different DS clusters can be used to estimate the cluster load. Use the `pmreadcounter` command as follows to see the value of the PLDB drop ratio:

```
SC_2_1# pmreadcounter | grep DropRatios | grep -i PLDB
```

The same command is used to check the drop ratio of a specific DS - the only difference is that instead of PLDB, the target DS is defined:

```
SC_2_1# pmreadcounter | grep DropRatios | grep -i DSx
```

In the above example, *x* stands for the target DS number.

In case the drop ratio is greater than or equal to 5 (that is, the amount of traffic drop is 5% or higher), contact the next level of support.

3.3.4 Checking Notifications Traffic

To check if the notifications traffic is being executed, check the following log located in the `syslog` of the payload blade or VM where the `CudbNotifications` process is running:

```
/var/log/PL_2_x/messages
```

To find the payload blade or VM running the `CudbNotifications` process, execute the following command:

```
CUDB3 SC_2_1# cudbHaState | grep NOTIF
```

The expected output must look similar as follows:

```
saAmfSISUHASState."safSu=PL-3,safSg=2N,safApp=ERIC-CUDB_SOAP_NOTIFIER"."safSi=2N-1": standby(2)
saAmfSISUHASState."safSu=PL-4,safSg=2N,safApp=ERIC-CUDB_SOAP_NOTIFIER"."safSi=2N-1": active(1)
```

Refer to *CUDB Notifications*, Reference [6] for more information.

3.3.5 Checking CPU Load

Use the `cudbMpstat` command to check the CPU load on all blades or VMs:

```
SC_2_1# cudbMpstat
```




The command displays the load results in tabular format. Refer to *CUDB Node Commands and Parameters*, Reference [5] for more information.

3.3.6 Checking Database Consistency

There are two methods to perform a consistency check:

Lightweight Consistency Check

Quick Lightweight Consistency Check performed through the command `cudbCheckConsistency`. It performs a quick check by comparing only the number of rows in the database tables of the PLDB/DSG master and slave replicas, with a short (sub-minute) execution time.

CUDB Consistency Check

The CUDB Consistency Check compares the contents of database tables containing Lightweight Directory Access Protocol (LDAP) entry attribute values, performing a deep check, that might need much more time than the former check, depending on the database utilization.

To perform a lightweight consistency check, use the `cudbCheckConsistency` command to check database consistency between database clusters (that is, between the master PLDB or DSG replicas and their slaves) as follows:

```
SC_2_1# cudbCheckConsistency
```

To run an in-depth consistency check on the data of LDAP entry attributes between database clusters (that is, between the master PLDB or DSG replicas and their slaves), use the `cudbConsistencyMgr` command as follows:

```
SC_2_1# cudbConsistencyMgr --order ms --node <nodeid>
{--dsg <dsgid> | --pl}
```

To find out how the CUDB Consistency Check function works and how it can be used, refer to the “Consistency Check” section of *CUDB Consistency Check*, Reference [8].





4 Problem Reporting

For any abnormal situation, refer to *CUDB Troubleshooting Guide*, Reference [7].

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

It is also important to collect the related data. For information on how to collect the data, refer to *Data Collection Guideline for CUDB*, Reference [9].





Glossary

For the terms, definitions, acronyms and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [2].





Reference List

CUDB Documents

- [1] *CUDB Users and Passwords*, 3/00651-HDA 104 03/10
- [2] *CUDB Glossary of Terms and Acronyms*
- [3] *Trademark Information*
- [4] *Typographic Conventions*
- [5] *CUDB Node Commands and Parameters*
- [6] *CUDB Notifications*
- [7] *CUDB Troubleshooting Guide*
- [8] *CUDB Consistency Check*
- [9] *Data Collection Guideline for CUDB*