CUDB Node Preventive Maintenance

APPLICATION INFORMATION

1/1541-CSH 109 067/10 Uen A

Contents

1 Introduction

This document describes the maintenance procedures to prevent breakdown and failures in Ericsson Centralized User Database (CUDB). This document provides the schedule of planned maintenance tasks to be performed to preserve and enhance CUDB system reliability. For more information on troubleshooting, refer to *CUDB Troubleshooting Guide*, Reference [1].

1.1 Scope

This document covers the following maintenance procedures:

- Continuous maintenance procedures.
- Daily maintenance procedures.
- Weekly maintenance procedures and tasks with longer periodicity.

1.2 Revision Information

Rev. A

This document is based on 1/1541-CSH 109 067/9 with the following changes:

- Updated hardware information.
- Virtualization deployment terminology updates throughout the document.
- Section 1.3 on page 1: Updated target groups and list of references.

1.3 Target Groups

This document is intended for Network Operations Center (NOC) personnel.

Procedures related to CUDB application are intended for users in the `cudbadmin` group. For more details on permissions and how to execute the related commands, refer to *CUDB Security and Privacy Management*, Reference [2] and *CUDB Node Commands and Parameters*, Reference [3].

1.4 Typographic Conventions

Typographic conventions can be found in the following document:

- *Typographic Conventions*

2 Overview

This document describes the following periodic checks and maintenance procedures to prevent breakdown and failures in CUDB:

- Continuous tasks and checks. It comprises all the monitoring activities that should be in place during all the time while the CUDB system is running in a production environment. It includes overseeing CUDB fault management and performance management data along with regular supervision of the network interconnecting all the CUDB nodes in the system.
- Daily tasks and checks. It is a comprehensive set of checks and management operations recommended to be executed on a daily basis in a live CUDB system.
- Weekly tasks and sporadic checks. It includes other minor management procedures and checks that can be scheduled weekly or less often.

The periodicity classification of preventive maintenance tasks is recommended to prevent failures in the CUDB system.

3 Continuous Maintenance Procedures

The following sections describe the maintenance procedures to be performed continually.

3.1 Supervising Alarms

CUDB nodes, as network elements, provide a northbound interface, used to communicate with Network Management Systems (NMSs), such as Operations and Systems Support (OSS). CUDB nodes send Simple Network Management Protocol (SNMP) traps to an external NMS for fault management through the Ericsson SNMP Agent (ESA) component. Performing a daily check on unacknowledged active alarms is recommended, even if the alarm status is continuously supervised. Alarms are issued in CUDB as SNMP traps sent from each CUDB node. Based on information displayed in alarms, the system administrator can locate the source of the event and by using the relevant documentation, steps can be taken to resolve or prevent failures.

CUDB node internal components have automatic procedures to detect faults and malfunctions.

To get a complete list of the raised alarms, execute the **`fmactivealarms`** command.

For more information on managing alarms, refer to *CUDB Node Fault Management Configuration Guide*, Reference [4].

The severity field, which is always set in all of the alarms, indicates the importance of the failure and its impact on normal CUDB system operation. It is recommended to handle alarms with higher severity values before alarms with lower priority values.

Non auto ceased alarms must be acknowledged once the cause that produced the alarm is completely solved. For more information on how to clear non auto ceased alarm, refer to *CUDB Node Fault Management Configuration Guide*, Reference [4].

An auto ceased alarm is terminated when the alarm with the cease of the faulty action is received.

3.2 Performance Management

CUDB includes a number of performance measurements to monitor CUDB node resources and system activity. Performance data is continuously collected and reported in eXtensible Markup Language (XML) files.

CUDB infrastructure and software components provide a set of statistic counters with valuable performance information that helps the system administrators to be aware of possible malfunctions in the system. Regularly check the values of counters for any deviation to baseline to detect and avoid inactivity or overload in any component of the system.

For further details on the counters and on how to read them, refer to *CUDB Performance Guide*, Reference [5].

Besides CUDB specific counters, CUDB allows applications whose data is hosted in CUDB to define their own counters. For more information on creating application counters, refer to *CUDB Application Counters*, Reference [6], and for a list and description on different counters, refer to *CUDB Counters List*, Reference [7].

3.3 Monitoring Backbone Connectivity

CUDB system relies on the Internet Protocol (IP) transport network of the customer to connect CUDB nodes placed in different network site locations. CUDB nodes implement some system monitoring functions and protocols to check availability and the ability to reach remote CUDB nodes in a system. Refer to the *System Level Availability* section of *CUDB High Availability*, Reference [8] for more information on these mechanisms.

Note: The backbone connectivity is essential for a healthy CUDB system, and therefore it is highly recommended to implement strong and exhaustive supervision mechanisms to check the connectivity between network sites hosting CUDB nodes to alleviate the impact of network failures on the system. For more information, refer to *CUDB Node Network Description*, Reference [9].

4 Daily Maintenance Procedures

The following sections describe maintenance procedures to be performed daily.

4.1 Create System Data Backup

The CUDB backup program utility allows the operator to perform system data backups to generate a complete backup of all data stored in a CUDB system.

Although CUDB is a highly resilient and redundant database system, it is strongly recommended to perform a system data backup everyday.

Backups can be scheduled or executed manually. For more information on how to perform CUDB system data backup, refer to *CUDB Backup and Restore Procedures*, Reference [10].

It is recommended to copy the generated backup files to an external equipment for extra data safety and to save disk storage system space.

4.2 Check Log Files

Check the high severity events from the log files daily. Filter the logs with the following severities:

- `EMERG` by executing the **`grep -i EMERG`** ***`<syslog file>`*** command.
- `ALERT` by executing the **`grep -i ALERT`** ***`<syslog file>`*** command.
- `CRIT` by executing the **`grep -i CRIT`** ***`<syslog file>`*** command.
- `ERR` by executing the **`grep -i ERR`** ***`<syslog file>`*** command.

The `EMERG`, `CRIT`, and `ERR` log files are stored on the System Controllers (SCs) in `/var/log/SC_2_1: messages, kernel, auth.` and `/var/log/SC_2_2: messages, kernel, auth.`, and on the payload blades or Virtual Machines (VMs) in `/var/log/PL*: messages_mysql, messages, kernel, auth.` while the `ALERT` log file is found in NDB logs and stored on the payload blade or VM in `/local/cudb/mysql/ndbd/data`.

For more information on extended CUDB logging, refer to *CUDB Node Logging Events*, Reference [11].

CUDB Logchecker is an additional software monitoring component on top of the current monitoring processes that aims to work as a preventive maintenance tool. For more information about CUDB Logchecker, refer to *CUDB Logchecker*, Reference [12].

CUDB Logchecker runs every 12 hours automatically. Following are the CUDB Logchecker scripts:

- `cudbGetLogs` script (to collect preventive maintenance logs for log analysis) runs at 00:25 and 12:25, unless it is configured otherwise.
- `cudbAnalyser` script (to analyze logs gathered and preprocessed by `cudbGetLogs`) runs at 00:50 and 12:50, unless it is configured otherwise.

The execution of CUDB Logchecker may produce alarms. Refer to the *Automatic Log Collection and Log Analysis* section of *CUDB Logchecker*, Reference [12] for more information.

4.3 Check Log Faults

The following sections describe the possible faults due to log analysis. These logs include both the manual checks explained in the previous chapter and logs generated by the CUDB Logchecker. If instructions for collecting and enclosing troubleshooting data due to a Customer Service Request (CSR) are needed when experiencing problems with the product, refer to *Data Collection Guideline for CUDB*, Reference [13] for more information.

4.3.1 cudbSystemStatus

Table 1 shows the `cudbSystemStatus` faults in different scenarios. For more information on `cudbSystemStatus`, refer to *CUDB Node Commands and Parameters*, Reference [3].

Table 1 cudbSystemStatus

Fault	Example Printout	Action

Process not running as expected.(1)	`[ERROR] cudbSystemStatus: CUDB Process is not running as expected (Severity: Warning) [-W-] CudbNotifications process.....`
	`...Not running in: PL0`
	
	Shows which CUDB process is not running as expected.
		Run the **`cudbSystemStatus -p`** command to check if the problem is resolved.
		
		If the printout still shows that a CUDB process is not running, contact the next level of support.

Table 1 cudbSystemStatus

Fault	Example Printout	Action

Replication delay of 1000-9999.(2)	`[ERROR] cudbSystemStatus: replication delay: 1000-9999 (Severity: Warning)`
	`Replication in DSG3(Node=246--Chan=1).. OK -- Delay = 1021`
	Shows which replication channel has a delay.
		Run the **`cudbSystemStatus -R`** command to verify that the replication is decreasing.
		If the replication is continuously increasing due to high load, consider reallocation.
		If the replication delay is not decreasing, contact the next level of support.

Replication delay > 10000.	`[ERROR] cudbSystemStatus: replication delay > 10000 (Severity: Minor)`
	`Replication in DSG3(Node=246--Chan=1).. OK -- Delay = 20001`
	Shows which replication channel has a delay.
		Run the **`cudbSystemStatus -R`** command to verify that the replication is decreasing.
		If the replication is continuously increasing due to high load, consider reallocation.
		If the replication delay is not decreasing, contact the next level of support.

BC server is not in a good shape.	`[ERROR] cudbSystemStatus: BC Server is not in a good shape. (Severity: Major)` `BC server in PL_2_5 ..... not running`	Run the **`cudbSystemStatus -b`** command to check if the problem still persists. If yes, contact the next level of support.

(1) Storage engine process can take up to one hour to restart.
(2) Replication delay is normal in the following situations: (a) Backbone issues or right after backbone issue (b) Backup restore (c) The affected PL or DS was in maintenance mode or unavailable.

4.3.2 NDB

Table 2 shows the NDB faults due to the execution of cudbAnalyser in different scenarios.

Table 2 NDB

Fault	Example Printout	Action
NDB cluster logs.	`[ERROR] NDB: Errors found in NDB cluster logs (Severity: Warning) > 2013-07-17 09:55:51 [MgmtSrvr] ALERT \ -- Node 3: Arbitration check won - node group majority`	If this printout comes without any other alarm or explanation for the last 12-hour monitoring period, contact the next level of support to investigate the logs.
ndb_out logs.	`[ERROR] NDB: Errors found in ndb_out logs (Severity: Warning) [ndbd] ALERT -- Node 4: Forced node shutdown completed. Caused by error 2303: 'System error, node killed during node restart by other node(Internal error, programming error or missing error message, please report a bug). Temporary error, restart node'.`	If this printout comes without any other alarm or explanation for the last 12-hour monitoring period, contact the next level of support to investigate the logs.
Database cluster hanging in phase 4.	`[ERROR] NDB: Mysql hanging in Phase 4 (Severity: Warning) > 2016-03-18 19:26:56 [ndbd] INFO -- refing dict lock to 4`	It is recommended to contact the next level of support to investigate the logs.

4.3.3 OS

Table 3 shows OS faults in different scenarios.

Table 3 OS

Fault	Example Printout	Action
High CPU load.(1)	`[ERROR] OS: high CPU load (Severity: Major) CUDB119 oam2 00:26am up 6 days 16:10, 0 users, load average: 25.35, 17.76, 14.16`	Log in to the affected blade or Virtual Machine (VM), and run the **`top`** command to check if the load is still high. If a software fault is suspected, contact the next level of support.

Table 3 OS

Fault	Example Printout	Action
procs_blocked > 9, indicating high load, or hanging IO.	`[ERROR] OS: Kernel printout: procs_blocked > 9, indicating high load, or hanging IO (Severity: Major) CUDB100 oam2 procs_blocked 10` This kernel printout shows which blade or VM is affected.	Check if there is another error printout by CUDB Logchecker that pinpoints an infrastructure or load related error. Log in to the affected blade or VM and run the **`dmesg`** command to see if the problem is caused by IO issues. Log in to the affected blade or VM and run the **`top`** command to see if the problem is caused by high load. It is recommended to contact the next level of support to investigate the issue.(2)
dmesg shows IO error or EXT3 filesystem error.	`[ERROR] OS: dmesg shows IO error or EXT3 filesystem error (Severity: Major) Feb 2 20:55:10 PL_2_5 kernel: end_request: I/O error, dev sda, sector 112313 Feb 2 20:55:10 PL_2_5 kernel: Buffer I/O error on device sda1, logical block 14039` Shows which blade or VM is affected.	It is recommended to contact the next level of support to investigate the issue.(2)
dmesg shows filesystem is mounted as read-only.	`[ERROR] OS: dmesg shows that filesystem is remounted as read-only (Severity: Major) Feb 2 20:55:15 PL_2_5 kernel: Remounting filesystem read-only` Shows which blade or VM is affected.	It is recommended to contact the next level of support to investigate the issue.(2)

Table 3 OS

Fault	Example Printout	Action

Network stat shows errors.	`[ERROR] OS: network stat shows errors (Severity: Minor)`
	`> CUDB124 oam1 eth3/statistics/rx_dropped 16467 K packets`
	`> CUDB124 oam1 bond0/statistics/rx_dropped 16467 K packets`
	`> CUDB124 oam2 eth3/statistics/rx_dropped 803 K packets`
	`> CUDB124 oam2 bond0/statistics/rx_dropped 803 K packets`
	Shows which blade or VM is affected.
		It is recommended to contact the next level of support to investigate the issue.(2)

In case the CUDB system is deployed on native BSP 8100 hardware, SMART (Self-Monitoring, Analysis and Reporting Technology) data indicates a possible imminent drive failure.(3)	`[ERROR] OS: SMART Health Status is NOT OK! (Severity: Major)`
	`CUDB136 oam1 SMART Health Status: HARDWARE IMPENDING FAILURE DATA ERROR RATE TOO HIGH [asc=5d, ascq=12]`
	`CUDB136 PL2 SMART Health Status: HARDWARE IMPENDING FAILURE DATA ERROR RATE TOO HIGH [asc=5d, ascq=12]`
		A drive in the affected blade shows a pre-failure condition. To avoid blade failure, the blade containing the drive must be replaced. For more information on blade replacement, refer to *Server Platform, Blade Replacement*, Reference [14].

Table 3 OS

Fault	Example Printout	Action
New core dumps were generated.	`[ERROR] OS: New core dumps were generated under /local2/dumps in the PLs and under /cluster/dumps in PL_2_5 (More details inside <cudb_pl_core_dumps> in log files) (Severity: Major)` `> -rw------- 1 root root 367984640 Feb 12 15:10 slapd.31928.PL_2_3.core`	It is recommended to contact the next level of support to investigate the issue.
New printouts regarding system Out of Memory Killer.	`[ERROR] OS: New printouts regarding system Out of Memory Killer (Severity: Warning)` `> May 10 17:37:05 PL_2_3 kernel: [191935.037336] Out of memory: Kill process 26499 (slapd) score 452 or sacrifice child`	If the printouts are still coming in the logs, contact the next level of support.

(1) High load for the ndbd process is normal. High load is present during manual LDAP import/export operations.
(2) This printout is never considered to be normal.
(3) In case of virtualized infrastructure, the Hardware Monitoring function should be used instead of the SMART function.

4.3.4 Config

Table 4 shows config fault in different scenarios.

Table 4 Config

Fault	Example Printout	Action

Error in custom cudb config checks.	`[ERROR] Config: Error in custom cudb config checks (Severity: Major)`
	`ERROR: cluster alarm -l -a alarm list is not empty`
	Shows which config checks have failed.
		Contact the next level of support to investigate the issue.

4.3.5 SWM

Table 5 shows SWM fault in different scenarios.

Table 5 SWM

Fault	Example Printout	Action
Multiple revisions from package: [package name].	`[ERROR] SWM: Multiple revisions from package (Severity: Minor) CUDB_NODE_CONFIG-CXP9015320`	Check **`cmw-repository-list`** on the node, and verify that the package shown in the printout has still more than one versions. Verify that there is no ongoing SW update or upgrade on the CUDB node, as the printout is normal in these cases. Contact the next level of support to investigate and resolve the issue.(1)

(1) Multiple revisions from one package in smf repository causes problems if an upgrade is attempted.

4.3.6 Database Cluster

Table 6 shows database cluster faults in different scenarios.

Table 6 Database Cluster

Fault	Example Printout	Action

Binlog is not written.	`[ERROR] MYSQL: Binlog is not written (Severity: Major)`
	`CUDB41 DS1_0 0`
	
	Shows the blade or VM where the binlog is not written.
		Run manual log collection and manual log analysis to verify that the problem is still present.(1) For more information about the procedure, refer to *CUDB Logchecker*, Reference [12].

Incorrect key file for table.	`[ERROR] MYSQL: Incorrect key file for table (Severity: Major)`
	`Sep 16 09:50:54 PL_2_5 mysqld: 110916 9:50:54 [ERROR] mysqld: Incorrect key file for table './mysql/ndb_binlog_index.MYI'; try to repair it`
	`Sep 16 09:50:54 PL_2_5 mysqld: 110916 9:50:54 [ERROR] mysqld: Incorrect key file for table './mysql/ndb_binlog_index.MYI'; try to repair it`
		If the printouts are still coming in the logs, contact the next level of support.

(1) This error printout is raised if the binlog is not written for the last 1440 minutes. This is normal for PLDB blades or VMs if there was no provisioning in the last 1440 minutes. In any other case, contact the next level of support.

4.3.7 System Monitor

Table 7 shows System Monitor faults in different scenarios.

Table 7 System Monitor

Fault	Example Printout	Action

System Monitor logs indicate there were major errors.	`[ERROR] System Monitor logs indicate there were major errors (Severity: Major)`
	`> Apr 19 18:21:33 SC_2_1 SM[23399]: INFO Auxiliar connection with site 2 has changed to LOST`
	`> Apr 19 18:21:34 SC_2_1 SM[23399]: INFO Connection status between SM leader in site 1 (S1-N202-I1) and BC service in site 2 has change to LOST`
		Run the **`cudbSystemStatus -b`** command to check if the problem still persists. If yes, contact the next level of support.

System Monitor logs indicate there were some warnings.	`[ERROR] System Monitor logs indicate there were some warnings [Towards site 2; Event count 14] (Severity: Warning)`
	`> Apr 20 01:10:25 SC_2_1 SM[23399]: INFO Auxiliar connection with site 2 has changed to SUSPENDED`
	`> Apr 20 01:10:26 SC_2_1 SM[23399]: INFO Auxiliar connection with site 2 has changed to RECONNECTED`
		Run the **`cudbSystemStatus -b`** command to check if the problem still persists. If yes, contact the next level of support.

4.3.8 BC Server

Table 8 shows BC Server faults in different scenarios.

Table 8 BC Server

Fault	Example Printout	Action
BC Server logs indicate there were minor errors.	`[ERROR] BC Server logs indicate there were minor errors (Severity: Minor) > Apr 20 09:25:37 SC_2_2 BC[myid:11]: INFO Got user-level KeeperException when processing sessionid:0xa52fbde985d104a type:ping cxid:0xfffffffffffffffe zxid:0xfffffffffffffffe txntype:unknown reqpath:n/a Error Path:null Error:KeeperErrorCode = Session moved`	It is recommended to contact the next level of support to investigate the issue.
BC Server logs indicate there were major errors.	`[ERROR] BC Server logs indicate there were major errors (Severity: Major) > Feb 18 13:17:46 SC_2_2 BC[myid:11]: ERROR Unexpected exception causing shutdown while sock still open > Feb 18 13:17:47 SC_2_2 BC[myid:11]: INFO Shutting down > Feb 18 13:17:47 SC_2_2 BC[myid:11]: INFO Shutdown called`	It is recommended to contact the next level of support to investigate the issue.

4.3.9 Master Unavailability

Table 9 shows Master Unavailability fault in different scenarios.

Table 9 Master Unavailability

Fault	Example Printout	Action

Master is not available for DSG.	`[ERROR] Error: Master is not available for DSG (Severity: Major)`
	`> Feb 12 14:45:06 SC_2_1 CS[14068]: Monitoring [ClusterSupervisor]: INFO - N203D0-6-BCClient: Enter getMasterStatus: masterless`
	`> Feb 12 14:45:06 SC_2_1 CS[14068]: Monitoring [ClusterSupervisor]: INFO - N203D0-6-BCClient: Information read in node /cudb/masterList/D0 -> masterless`
		Run the **`cudbSystemStatus -R`** command to check if the problem still persists. If yes, contact the next level of support.

4.3.10 Software Platform

Table 10 shows Software Platform faults in different scenarios.

Table 10 Software Platform

Fault	Example Printout	Action
Not all SU have Operational State Enabled.	`[ERROR] SAF: Not all SU have Operational State Enabled (Severity: Warning) saAmfNodeOperState."safAmfNode=SC-1,safAmfCluster=myAmfCluster": Disabled`	Run the **`cudbHaState`** command to check if the problem still persists. If yes, contact the next level of support.
There are unassigned HA states.	`[ERROR] SAF: there are unassigned HA states (Severity: Warning) saAmfSISUHAState."safSu=SC-1,safSg=NoRed-PMCounter,safApp=ERIC-LDE"."safSi=NoRed8": unassigned(3)`	Run the **`cudbHaState`** command to check if the problem still persists. If yes, contact the next level of support.

Table 10 Software Platform

Fault	Example Printout	Action
One or more nodes are locked in the cluster.	`[ERROR] SAF: One or more node is locked in the cluster (Severity: Warning) saAmfNodeAdminState."safAmfNode=SC-1,safAmfCluster=myAmfCluster": Locked`	Run the **`cudbHaState`** command to check if the problem still persists. If yes, contact the next level of support.
There are active cluster alarms.	`[ERROR] SAF: There are active cluster alarms (Severity: Major) Node Hostname Severity Type Problem Information 1 SC_2_1 Major 2 Ethernet Bonding Bonding degraded on bond0 (link down on eth4)`	Run the **`cluster alarm -a -l`** command to check if the problem still persists. If yes, contact the next level of supports.

4.4 Perform Lightweight Consistency Check

It is recommended to perform a Lightweight Consistency Check as described in the *Detecting Inconsistencies between Replicas* section of *CUDB Data Storage Handling*, Reference [15].

4.5 Check Infrastructure Related Components (for BSP 8100 only)

Before starting the infrastructure check, perform Lightweight Directory Access Protocol (LDAP) log error checks.

Refer to the ‘‘BSP Fault Management’’ and ‘‘BSP System Notifications’’ documents in the BSP 8100 CPI for detailed information on how to supervise and take preventive actions on the BSP 8100 infrastructure.

4.6 Check Operating System

CUDB nodes use Linux Distribution Extension (LDE) as operating system. For more information on LDE, refer to *LDE Management Guide*, Reference [19].

Perform the following checks in each blade or VM:

- Check the available free memory by executing the **`> free`** command.
- Check the disk storage system utilization by executing the **`> df -k`** command.

- Check CPU utilization by executing the `> mpstat -P ALL 1 5` command.

In case of abnormal repeated values on any of these measurements along with performance management data deviations, please contact Ericsson Support.

5 Weekly Maintenance Procedures and Tasks with Longer Periodicity

The following sections describe weekly maintenance procedures and tasks.

5.1 Creating Software and Configuration Backups

It is recommended to have backups on software packages and configuration files periodically.

For more information on the backup and restore procedure, refer to *CUDB Backup and Restore Procedures*, Reference [10].

5.2 User Administration

For information on user administration, refer to *CUDB System Administrator Guide*, Reference [16].

5.3 Performing Consistency Check

It is recommended to perform a Consistency Check as described in *CUDB Consistency Check*, Reference [17].

Glossary

For the terms, definitions, acronyms, and abbreviations used in this document, refer to *CUDB Glossary of Terms and Acronyms*, Reference [18].

Reference List

CUDB Documents

[1] *CUDB Troubleshooting Guide*

[2] *CUDB Security and Privacy Management*

[3] *CUDB Node Commands and Parameters*

[4] *CUDB Node Fault Management Configuration Guide*

[5] *CUDB Performance Guide*

[6] *CUDB Application Counters*

[7] *CUDB Counters List*

[8] *CUDB High Availability*

[9] *CUDB Node Network Description*

[10] *CUDB Backup and Restore Procedures*

[11] *CUDB Node Logging Events*

[12] *CUDB Logchecker*

[13] *Data Collection Guideline for CUDB*

[14] *Server Platform, Blade Replacement*

[15] *CUDB Data Storage Handling*

[16] *CUDB System Administrator Guide*

[17] *CUDB Consistency Check*

[18] *CUDB Glossary of Terms and Acronyms*

Other Ericsson Documents

[19] *LDE Management Guide*