

LDE Management Guide

Linux Distribution Extensions

USER GUIDE

Copyright

© Ericsson AB 2015 - 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

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

RHEL is the registered trademark of Red Hat, Inc. in the United States and other countries.

All trademarks are the property of their respective owners.



Contents

1	Introduction	1
2	Prerequisites	3
3	Related Information	5
4	Revision Information	7
5	System Configuration	9
5.1	Overview	9
5.2	The lde-config and cluster.conf	9
5.3	The Etc-Overlay-Framework	40
6	Kickstart Management (for LDEfR)	51
6.1	Overview	51
6.2	Creating Kickstart structure for a blade	51
7	Software Configuration (for LDEfS and LDEfR)	53
7.1	Overview	53
7.2	Repository Management	53
7.3	Software Management	55
7.4	Upgrading To New System Version	58
8	Software Configuration (for LDEwS)	61
8.1	Adding an Application Package	61
8.2	Removing an Application Package	61
8.3	Upgrading an Application Package	62
8.4	Upgrading an LDEwS Package	62
8.5	Listing Packages	64
8.6	Synchronizing Packages	65
8.7	Activating Packages	65
9	User and Group Management	67
9.1	Overview	67
9.2	Adding/Removing/Modifying Users and Groups	67
9.3	Make a User or Group Global	67
9.4	Changing Password	67



9.5	Account and Password Aging	68
9.6	Forced Password Change	68
9.7	Allowing Login Users (for LDEwS)	69
9.8	Login Information	69
9.9	Inactivity Logout	69
10	Logging In Remotely	71
10.1	SSH	71
11	Backup	73
11.1	Overview	73
11.2	Creating a Backup	73
11.3	Restoring a Backup	73
12	Local Snapshot	75
12.1	Overview	75
12.2	Creating a Local Snapshot on LDEfS	75
12.3	Creating a Local Snapshot on LDEwS	75
12.4	Restoring a Local Snapshot on LDEfS	76
12.5	Restoring a Local Snapshot on LDEwS	76
12.6	Delete a Local Snapshot on LDEfS	77
12.7	BRF (for LDEwS)	77
13	Reboot	85
13.1	Overview	85
13.2	Reboot a Single blade	85
13.3	Reboot the Whole Cluster	85
13.4	Bypass the BIOS When Rebooting	85
14	Power	87
14.1	Overview	87
14.2	Configuration	87
14.3	Power On	87
14.4	Power Off	87
14.5	Power Reset	87
14.6	Power Status	88
15	Alarms	89
15.1	Overview	89
15.2	Alarm Status	89



16	Configuration Management (for LDEwS)	91
16.1	Example	92
17	Cluster tool	95
18	Network Services	97
18.1	Listening Services	97
18.2	List Services	98
18.3	SSH server	98
19	Preventive Maintenance	99
19.1	System Check	99
19.2	Logs	100
19.3	Copying large files to shared replicated storage	100
19.4	Maintenance Mode	101
19.5	Trouble Reporting	101
	Reference List	103





1 Introduction

This document describes how to configure and manage a system that has installed Linux Distribution Extensions (LDE).

Scope

This document covers the following topics:

- Configuration
- Management

Target Groups

This document is intended as an introduction for personnel involved in any operation and maintenance activity of LDE.





2 Prerequisites

Before preventative maintenance can be performed (see Section 19 on page 99), a system administrator, operator or maintenance technician must have access to all relevant information about the system such as IP addresses and login credentials and be familiar with the following:

- Linux
- IP based networks
- LDE systems documentation (or at least have access to the document set for future reference)





3 Related Information

The definition and explanation of acronyms and terminology, information about trademarks used, and typographic conventions can be found in the following documents:

- *LDE Glossary of Terms and Acronyms*, Reference [7]
- *LDE Trademark Information*, Reference [8]
- *Typographic Conventions*, Reference [10]





4 Revision Information

Other than editorial changes, this document has been revised to revision A according to the following:

- Migration to LDE 4.0 product number





5 System Configuration

5.1 Overview

The system is configured either using a plain text file, stored as `/cluster/etc/cluster.conf` or as static changes to configuration files using the `etc-overlay-framework`. The `cluster.conf` file contains blade and site specific parameters, such as number of included blades, IP addresses, external Domain Name System (DNS) / Network Time Protocol (NTP) servers while the `etc-overlay-framework` is typically used for very application specific configuration such as hardening.

The `cluster.conf` is applied on the system using the `lde-config` command while the `etc-overlay-framework` is applied by installing an RPM. For more information about this, see Section 5.2.4.2 Reload on page 38.

5.2 The lde-config and cluster.conf

5.2.1 Legend

The symbols and expressions used in this document when describing the available configuration parameters are shown in Table 1.

Table 1 Legend

Symbols and Expressions	Description
<code><id> : <number></code>	<code><id></code> is a number between 1 and 254 that uniquely identifies a blade within the system.
<code><name> : <alpha num></code>	<p><code><name></code> is a string consisting of a number between 0 and 9, lower and capital characters a to z, “.” and “-”.</p> <p>The name is linked to a <code>nodegroup</code>.</p>



Table 1 Legend

Symbols and Expressions	Description
<code><role> : { control payload }</code>	<p>Defines the role of a blade in the system. The following two roles are supported:</p> <ul style="list-style-type: none">• <code>control</code> – A control blade boots from disk and runs services to support the rest of the blades in the system. Max 2 control blades are allowed within a system.• <code>payload</code> – A payload blade boots over the network and is dependent upon services provided by the control blades.
<code><target> : { <id> <role> <node group> all }</code>	<p>Defines to which blade or blades a specific configuration is to be applied. The target can be specified in the following four ways:</p> <ul style="list-style-type: none">• By node ID.• By blade role.• By node group, see the <code>nodegroup cluster.conf</code> parameter.• All blades.

5.2.2 Parameters

The parameters are shown in Table 2 until Table 39.



Table 2 Alarm

Syntax	<code>alarm <target> <type> <threshold></code>	
Description	<p>Specifies the threshold for an alarm.</p> <p>Without this configuration the disk usage and memory usage alarms use a default threshold of 90%.</p>	
Options	<code><target></code>	Target node(s).
	<code><type></code>	<p>Alarm type. Can be one of the following:</p> <ul style="list-style-type: none"> • <code>disk_usage</code> • <code>disk_usage_minor</code> • <code>disk_usage_major</code> • <code>disk_usage_critical</code> • <code>memory_usage</code> <p>The user can set a threshold for different severity levels of disk usage. The types <code>disk_usage</code> and <code>disk_usage_major</code> are on the same severity level but if both are present <code>disk_usage_major</code> overrides <code>disk_usage</code>.</p>
	<code><threshold></code>	Threshold for the alarm (in %). See the Operating Instruction (OPI) of the alarm in question for more information.
Examples	<pre>alarm control disk_usage_minor 60 alarm control disk_usage_major 80 alarm control disk_usage_critical 90 alarm 3 memory_usage 70 alarm control memory_usage 50</pre>	



Table 3 Bonding

Syntax	bonding <target> <interface> <parameter> <parameter value>	
Description	<p>Set driver parameters for the specified bonding interface. The bonding mode should be set at the interface definition (see the 'interface' command).</p> <p>For a specific node-interface pair, use only one of the monitoring modes' parameters. E.g.: 'arp_ip_target' and 'miimon' should not be used together on the same interface.</p> <p>For a comprehensive documentation about the bonding driver parameters, see: https://www.kernel.org/doc/Documentation/networking/bonding.txt</p> <p>On the internal network, currently only the "arp" parameter is supported.</p>	
Options	<target>	Target node(s).
	<interface>	Bonded interface.
ARP monitoring parameters		



Table 3 Bonding

Parameter s	<i>parameter</i>	<i>parameter value</i>	
	{ arp arp_ip_target }	<ip address 1> ... <ip address n>	<p>Switches to Address Resolution Protocol (ARP) monitoring for a bonding interface and specifies one or more ARP targets for the monitoring of this interface.</p> <p>At least two IP addresses, specifying different ARP targets, must be used to avoid having a single point of failure.</p> <p>Whether to use ARP monitoring or link state monitoring (which is default) is highly dependant on the setup in general. For example, some switches may not report changes in the link state quick enough for link state monitoring to work (in which case ARP monitoring is preferred). Other switches may stop forwarding packets while still answering the ARP (in which case Media Independent Interface (MII) monitoring is preferred). The best way to see which is better suited for a certain environment is to verify the functionality using fail over tests.</p>
	arp_interval	<milliseconds>	<p>Specifies the ARP link monitoring frequency in milliseconds.</p> <p>The default value is: 500</p>
	arp_validate	{ 0 1 2 3 }	<p>Specifies whether or not ARP probes and replies should be validated in the active-backup mode.</p> <p>The default value is: 3</p>
MII monitoring parameters			



Table 3 Bonding

Parameter s	<i>parameter</i>	<i>parameter value</i>	
	miimon	<i><milliseconds></i>	Specifies the MII link monitoring frequency in milliseconds. The default value is: 100
	updelay	<i><milliseconds></i>	Specifies the time, in milliseconds, to wait before enabling a slave after a link recovery has been detected. The default value is: 60000
	downdelay	<i><milliseconds></i>	Specifies the time, in milliseconds, to wait before disabling a slave after a link failure has been detected. The default value is: 0
	use_carrier	{ 0 1 }	Specifies whether or not miimon should use MII or ETHTOOL ioctls vs. netif_carrier_ok() to determine the link status. The default value is: 1
<i>Additional supported parameters</i>			



Table 3 Bonding

Parameter s	<i>parameter</i>	<i>parameter value</i>	
	num_grat_arp	0 - 255	Specify the number of peer notifications (gratuitous ARPs and unsolicited IPv6 Neighbor Advertisements) to be issued after a failover event. The default value is: 1
	primary	<interface>	Specifies which slave is the primary device to use. The default value is <if0>. (See the interface command)
	primary_reselect	{ 0 1 2 }	Specifies the reselection policy for the primary slave. The default value is: 0
	lacp_rate	{ 0 1 }	Option specifying the rate in which we'll ask our link partner to transmit LACPDU packets in 802.3ad mode. The default value is: 0
	ad_select	{ 0 1 2 }	Specifies the 802.3ad aggregation selection logic to use. The default value is: 0
	xmit_hash_policy	{ 0 1 2 }	Selects the transmit hash policy to use for slave selection in balance-xor and 802.3ad modes. The default value is: 0
	fail_over_mac	{ 0 1 2 }	Specifies whether active-backup mode should set all slaves to the same MAC address at enslavement (the traditional behavior), or, when enabled, perform special handling of the bond's MAC address in accordance with the selected policy. The default value is: 0
	all_slaves_active	{ 0 1 }	Specifies that duplicate frames (received on inactive ports) should be dropped (0) or delivered (1). The default value is: 0
	resend_igmp	0 - 255	Specifies the number of IGMP membership reports to be issued after a failover event. One membership report is issued immediately after the failover, subsequent packets are sent in each 200ms interval. The default value is: 1



Table 3 Bonding

<i>Unsupported parameters</i>			
Parameter s	<i>parameter</i>	<i>parameter value</i>	
	num_unsol_na		Use 'num_grat_arp' to set this parameter's value. The default value is: 1
	max_bonds		Set to the driver's default value allowing for multiple bonds.
	tx_queues		Set to the driver's default value: 16
Examples	<pre>bonding all bond0 arp 192.168.0.101 192.168.0.102 bonding payload bond1 arp_ip_target 192.168.0.101 192.168.0.102 192.168.0.103 bonding all bond1 miimon 200 bonding 3 bond0 num_grat_arp 10 bonding 1 bond0 primary eth1</pre>		

Table 4 Boot

Syntax	boot <ip address>	
Description	Defines the IP address of the boot server that all blades should use. This parameter can be defined multiple times if more than one boot server (running the Trivial File Transfer Protocol (TFTP) service) is used. Note: It is recommended to use two boot servers to improve operational performance.	
Options	<ip address>	IP address to use. Only IPv4 addresses are accepted.
Examples	boot 192.168.0.200	

Table 5 Bootserver



Table 5 Bootserver

Syntax	<code>bootserver <bootserver name> <attributes> <attribute value></code>		
Description	Defines attributes for the bootserver property		
Attributes	<i>attributes</i>	<i>attribute value</i>	
	mode	slave	slave - run only a dhcp server and be slave to another bootserver that provides the tftp server (external or internal). Requires the attribute master to be set for the bootserver.
		shared	shared - run dhcp and tftp servers on all blades with this property (should be limited to 2). Requires the bootserver to have two mip attributes per serving network
		backup	run only one dhcp and tftp server (second blade used if primary fails). Requires the bootserver to have one mip attribute per serving network
	servingblades	<i><group, list of blades></i>	Defines how the bootserver will be segmented. Can be set to 'all' to indicate a bootserver should serve all blades
	network	<i><network name></i>	Defines which network the boot server is serving. Can be defined several times if the boot server should serve several networks
	mip	<i><mip name></i>	Defines which mip the boot server should use to serve tftp. Can be defined several times to support both several networks but also to support the shared mode.
	master	<i><ip></i>	Defines the tftp server ip when running the boot server in slave mode.
Examples	<pre> network subA 192.168.69.0/24 network subB 192.168.70.0/24 mip control subA mip eth3:1 subA 192.168.69.130 mip control subB mip eth4:1 subB 192.168.70.130 property control bootserver boots bootserver boots mode backup bootserver boots servingblades all </pre>		



Table 6 Coredump

Syntax	coredump <target> <size> coredump <target> unlimited	
Description	Specifies the default core dump size for processes.	
Options	<target>	Target node(s).
	<size>	The maximum size of the core dump in Kilobytes.
	unlimited	The theoretical unlimited size of a core dump.
Examples	coredump all 1000000 coredump 1 unlimited	

Table 7 Default-output

Syntax	default-output <value>	
Description	Specifies if the console is to default to a serial or Video Graphics Adapter (VGA) output for the whole cluster.	
Options	<value>	If set to <code>serial</code> the output is sent to a serial console. If set to <code>vga</code> the output is sent to a VGA console.
Examples	default-output serial default-output vga	

Table 8 Disable-serial (for LDEwS)

Syntax	disable-serial <value>	
Description	Disables the serial console for the whole cluster. In contrast to <code>default-output</code> , which sets what the system is to default to, <code>disable-serial</code> removes the serial console option completely.	
Options	<value>	Either <code>on</code> to use serial console or <code>off</code> to disable serial console.
Examples	disable-serial on disable-serial off	



Table 9 DNS

Syntax	dns <target> <ip address>	
Description	Specifies an external DNS server. This parameter can be specified multiple times if multiple external DNS servers are used.	
Options	<target>	Target blade(s).
	<ip address>	IP address of DNS server.
Examples	dns all 163.168.48.202 dns payload 163.168.48.203	

Table 10 HAIAD (High Availability Internet Address Daemon Control)

Syntax	haiad <target> network <network name> ip <ip address> dev <if0> [<if1> [<ifN>]]		
Description	Defines a HAIAD monitored network. Note: Separate boot networks are required on the physical interfaces.		
Options	option	value	
		<target>	Target node(s).
	network	<network name>	Name of network to be monitored by HAIAD.
	ip	<ip address>	IP-address for target or “dynamic” for dynamic assignment.
	dev	<if0> [<if1> [<ifN>]]	Physical interfaces to run HAIAD over.
Examples	haiad all network internal ip dynamic dev eth0 eth1		

Table 11 Host

Syntax	host <target> <ip address> <name>	
Description	Defines a host entry in the /etc/hosts file. This parameter can be defined multiple times if multiple host entries should be added.	
Options	<target>	Target blade(s).
	<ip address>	IP address of the host.
	<name>	Name of the host.
Examples	host all 138.122.45.19 mgrserver host control 145.131.87.98 oam	



Table 12 Interface



Table 12 Interface

Syntax	<pre> interface <target> <device> ethernet <mac> interface <target> <device> bonding <if0> <if1> interface <target> <device> bonding <if0> <if1> mode <bonding mode> interface <target> <device>:<alias id> alias interface <target> <device>.<vlan id> vlan interface <target> <device> macvlan <if> [<mac>] </pre>	
Description	Defines a network interface for a node, either a physical, bonding, alias, Virtual Local Area Network (VLAN) or Media Access Control (MAC) VLAN interface.	
Option	<target>	Target node(s).
	<device>	Device name of interface.
	<mac>	Physical address of device. If the <mac> field is supplied for a MAC VLAN interface, where <target> is a group of nodes (such as control, all, or a node group), the node ID is added to the last three octets of the MAC address, which must have room for all nodes.
	<if0> <if1>	Interfaces to bond together.
	<bonding mode>	<p>The bonding mode to use on the bonding interface. Supported values are: 0 - 6.</p> <p>If omitted, bonding mode is set to mode 1 (active-backup) as default.</p> <p>Currently this option is not supported on the internal network.</p>
	<alias id>	Unique alias ID for interface.
	<vlan id>	Unique VLAN ID for interface. The maximum allowed value of a VLAN ID is 4094.
	<if>	Physical interface to use as MAC VLAN base.
Examples	<pre> interface 1 eth0 ethernet 00:02:B3:B8:AC:01 interface all bond0 bonding eth0 eth1 interface all bond1 bonding eth2 eth3 mode 0 interface control bond0:1 alias interface payload bond0.10 vlan interface all mv10 macvlan eth0 interface 1 mv11 macvlan eth1 52:03:12:A7:57:01 interface all mv12 macvlan eth2 52:03:12:A7:56:00 </pre>	



Table 13 IP

Syntax	<code>ip <target> <interface> <network> <ip address></code>	
	<code>ip <target> <interface> <network> dynamic</code>	
Description	Defines an IP address for an interface. When <code>dynamic</code> is used, the IP address will be automatically assigned based on the node ID. For example, if node 2 wants an address on a network with address/prefix <code>192.168.0.0/24</code> , it will be given the IP address <code>192.168.0.2</code> .	
Options	<code><target></code>	Target node(s).
	<code><interface></code>	Interface the IP address is to be assigned to.
	<code><network></code>	Network the IP address belongs to.
	<code><ip address></code>	IP address. Both IPv4 and IPv6 addresses are accepted.
Examples	<code>ip all bond0 internal dynamic</code> <code>ip 3 eth2 traffic 192.168.2.100</code>	



Table 14 IPMI

Syntax	<code>ipmi <target> <ip address> <user></code>	
Description	<p>Defines power management using Intelligent Platform Management Interface (IPMI).</p> <p>In a test system, password can be defined as last parameter to <code>ipmi</code>, and this will then be used for authentication towards the IPMI controller. However, the password should never be defined in a live system. Instead the password will be asked for every time a IPMI related command is issued.</p> <p>Note that remote power management require the system to support IPMI over LAN.</p>	
Options	<code><target></code>	Target blade(s).
	<code><ip address></code>	<p>IP address to reach the IPMI controller of the blade. Must be coordinated with IPMI configuration made in BIOS of the blade.</p> <p>If the target is <code>all</code>, <code>control</code> or <code>payload</code>, the address is treated as a base address and the node ID is used as an offset to the address. For example, if the address is <code>192.168.0.200</code>, then blade 1 should have IP address <code>192.168.0.201</code>. If instead the target is a specific node ID, the address should be the actual IP address of that blade.</p> <p>Note: The IPMI address must be within the same network defined as the <code>internal</code> network.</p>
	<code><user></code>	User to authenticate towards the IPMI controller. Must be coordinated with the IPMI configuration made in e.g. BIOS of the blade.
Examples	<pre>ipmi all 192.168.0.200 impiusr ipmi 2 192.168.0.281 root</pre>	

Table 15 *IPtables*

Syntax	<code>iptables <target> <command></code>	
Description	Defines a rule in <code>iptables</code> . Rules will be run in the order specified in this configuration.	
Options	<code><target></code>	Target blade(s).
	<code><command></code>	Specifies the parameters that should be passed to <code>iptables</code> . This can be any parameter accepted by <code>iptables</code> , please see man page of <code>iptables(8)</code> for more information.
Examples	<p>On all nodes, drop packets destined from source address 10.0.0.1:</p> <pre>iptables all -A INPUT -s 10.0.0.1 -j DROP</pre> <p>On all nodes, accept SSH traffic destined for the 192.168.0.0/24 network and drop all other SSH traffic:</p> <pre>iptables all -A INPUT -p tcp --dport 22 -d 192.168.0.0/24 -j ACCEPT</pre> <pre>iptables all -A INPUT -p tcp --dport 22 -j DROP</pre>	

Table 16 *IP6tables*

Syntax	<code>ip6tables <target> <command></code>	
Description	Defines a rule in <code>ip6tables</code> . Rules will be run in the order specified in this configuration.	
Options	<code><target></code>	Target node(s).
	<code><command></code>	Specifies the parameters that are to be passed to <code>ip6tables</code> . This can be any parameter accepted by <code>ip6tables</code> . For more information, see the man page of <code>ip6tables(8)</code> .
Examples	<pre>ip6tables all -A INPUT -s fe80::21f:29ff:fe04:f9fa -j DROP</pre>	



Table 17 Kernel-cmdline

Syntax	<code>kernel-cmdline <target> <command line parameters></code>	
Description	Defines the command line parameters to be given to the kernel on a blade in the cluster. Note: This option must be used with care and should only be used under advice from Ericsson support.	
Options	<code><target></code>	Target blade(s).
	<code><command line parameters></code>	The command line parameters required to be given to the kernel.
Examples	<code>kernel-cmdline all noirqbalance</code>	

Table 18 Keymap

Syntax	<code>keymap <target> <keymap></code>	
Description	Defines the keymap to be used on a blade in the cluster. If no keymap is specified, the standard US keymap is used.	
Options	<code><target></code>	Target blade(s).
	<code><keymap></code>	The keymap to use. Examples include <code>se-latin1</code> and <code>it</code> . See the directory <code>/usr/share/kbd/keymaps/i386/</code> or <code>/lib/kbd/keymaps/i386/qwerty/</code> (depending on Linux distribution) on a running blade for all available keymaps. Remove <code>.map.gz</code> from the keymap name.
Examples	<code>keymap all se-latin1</code>	



Table 19 Loader (for LDEwS)

Syntax	loader <target> <target 1> ... <target n>	
Description	The target of the loader group will act as loader for the rest of the defined node(s). This makes it possible to partition the cluster into sections reducing load on shared resources during a cluster start/restart.	
Options	<target>	Target node, acting as loader for the members of this group.
	<target 1> ... <target n>	One or more members of the loading group.
Examples	Cluster consisting of two loading groups, where the first is served by node 1 and 2, and the second serviced by node 4 and 5. loader control payload loader control 3 4 5 loader 4 6 7 8 loader 5 6 7 8	
Exceptions	Ignored in single node installations.	

Table 20 MIP

Syntax	mip <target> <name> <interface> <network> <ip address>	
Description	Defines a movable IP address and associates it with a symbolic name. This IP address can later be activated/deactivated at runtime by referring to its name.	
Options	<target>	Target blade(s).
	<name>	Symbolic name to be associated with the movable IP address.
	<interface>	Interface the IP address should be applied.
	<network>	Network the IP address belongs to.
	<ip address>	IP address.
Examples	mip control nfs bond0:1 internal 192.168.0.100 mip control oam eth2.4:1 nbi 192.168.1.100	



Table 21 Netconsole

Syntax	netconsole <target> <interface> <destination address> <destination port> [<destination mac>]	
Description	Defines a network console destination. All console messages will be sent to the destination address and port as plain-text UDP traffic.	
Options	<target>	Target blade(s), which the rule applies to
	<interface>	Network interface that is to be used.
	<destination address>	The address of the destination.
	<destination port>	The port of the destination.
	<destination mac> (optional)	The MAC address of the destination. This can be left out, but will require that the ARP handling in the kernel is functional in order for the packages to be sent. The value <code>auto</code> will try to fetch the MAC address of the target during boot and then set this as the target MAC. Note that the MAC address is only used on the local IP network.
Examples	netconsole all eth0 192.168.0.254 1234 netconsole 1 eth2 10.3.14.42 5678 auto netconsole 2 eth5 130.100.96.239 9012 18:a9:05:bf:b8:48	



Table 22 Network

Syntax	<code>network <name> <network address>/<network prefix></code>	
Description	<p>Defines a network with a name together with its network address and prefix.</p> <p>An extra feature provided here is that using a <code><network address></code> with an offset, such as <code>192.168.0.15</code> combined with using <code>ip <target> <network prefix> <interface> dynamic</code> the dynamic address range will start from the offset, i.e. the address for node 1 will be <code>192.168.0.16</code>.</p> <p>The network name <code>internal</code> has a special meaning and is to be used to specify the cluster's internal network.</p>	
Options	<code><name></code>	Symbolic name of the network.
	<code><network address></code>	The address of the network. Both IPv4 and IPv6 addresses are accepted. Providing a network address with an offset into the network will start dynamic ip address assignment from this offset.
	<code><network prefix></code>	Prefix of the network.
Examples	<pre>network internal 192.168.0.0/24 network default 0.0.0.0/0</pre>	

Table 23 NFS

Syntax	<code>nfs <ip address> [<export path>]</code>	
Description	Defines the IP address of the NFS server that all blades should use.	
Options	<code><ip address></code>	IP address to use.
	<code><export path> (optional)</code>	The export path from the NFS server that should be mounted under <code>/cluster</code>
Examples	<pre>nfs 192.168.0.100</pre>	
Exceptions	Ignored in single blade installations.	



Table 24 Node

Syntax	<code>node <id> control <hostname></code> <code>node <id> payload <hostname></code>	
Description	Defines a blade in the cluster and gives it a role and an identity, either a control blade or a payload blade.	
Options	<code><id></code>	ID of the blade.
	<code><hostname></code>	Host name of the blade. If no host name is defined for the blade, it will be assigned as <code>node<id></code> .
Examples	<code>node 1 control</code> <code>node 3 payload PL3</code>	
Exceptions	Additional blades ignored in single blade installations.	

Table 25 Nodegroup

Syntax	<code>nodegroup <name></code> <code>nodegroup <name> <target 1> ... <target n></code>	
Description	Defines a node group in the cluster and assigns members to it. Node groups consist of targets as defined above, and can be used where targets are to be supplied. Node groups can be empty for the purpose of, for example, preparing for future expansion of the cluster.	
Options	<code><name></code>	Name of the node group.
	<code><target 1> ... <target n></code>	The id(s), role(s), or node group(s) that will be added to the node group.
Examples	<code>nodegroup service 10 11</code> <code>nodegroup loaders control service 20 21</code>	

Table 26 NTP

Syntax	<code>ntp <address></code>	
Description	Defines an external NTP server. This parameter can be defined multiple times if more than one external NTP server is used. Note: It is recommended to use three or more external NTP servers to improve operational performance and fault tolerance.	
Options	<code><address></code>	IP address or host name of NTP server
Examples	<code>ntp 134.168.48.200</code> <code>ntp time.mydomain.com</code>	

Table 27 *Property*

Syntax	<code>property <target> <property name></code>	
Description	Assigns a specific property to the target node(s)	
Options	<code><target></code>	Target node(s).
Properties	<code>bootserver</code>	Start boot services on the target blade.
Examples	<code>property 3 bootserver boot_external_blade</code>	

Table 28 *Quick-reboot*

Syntax	<code>quick-reboot <target> <value></code>	
Description	Specifies if a node or nodes are to use quick reboot or not.	
Options	<code><target></code>	Target node(s).
	<code><value></code>	Either <code>on</code> (default) to enable quick reboot or <code>off</code> to disable quick reboot.
Examples	<code>quick-reboot control off</code> <code>quick-reboot payload on</code>	

Table 29 *RAM-rootfs-size (for LDEwS)*

Syntax	<code>ram-rootfs-size <target> <size></code>	
Description	Configures the size of the root filesystem in RAM. Not applicable for blades running with the <code>disk_cache</code> installation option.	
Options	<code><target></code>	Target blade(s).
	<code><size></code>	The desired size of the root filesystem (MB).
Examples	<code>ram-rootfs-size all 3072</code>	
Exceptions	Ignored for blades running with disk caches	



Table 30 Route

Syntax	<pre>route <target> <network> interface <interface> [source <source address>]</pre> <pre>route <target> <network> gateway <address> [source <source address>]</pre>	
Description	Defines a IP network route.	
Options	<target>	Target blade(s).
	<network>	Name of the network that should be routed.
	<interface>	Name of the interface to route through packets.
	<address>	IP address or host name of the gateway.
	<source address>	IP source address that should be set when using the route.
Examples	<pre>route control default gateway 145.210.123.1</pre> <pre>route all traffic interface eth2</pre> <pre>route control default gateway 10.10.23.45 source 192.168.0.23</pre>	

Table 31 *Scaling*

Syntax	<pre>scaling [bootserver <bootserver>] [network <network>] <IP from> <IP to> scaling [bootserver <bootserver>] [network <network>] leasetime <seconds></pre>	
Description	Configures the DHCP server to enable scaling.	
Options	<bootserver>	A boot server that will issue temporary IP(s) to scaled blade(s).
	<network>	Name of the network of the boot server issuing temporary addresses.
	<IP from>	Inclusive start of the temporary IP address pool.
	<IP to>	Inclusive end of the temporary IP address pool.
	<seconds>	Value in seconds the DHCP server will keep a temporary IP address lease. Default is 300 seconds.
Examples	<pre>scaling 192.168.0.200 192.168.0.230 scaling leasetime 600 scaling network scaling_net 192.168.0.200 192.168.0.230 scaling bootserver scaling_dhcp network scaling_net 192.168.0.200 192.168.0.230 scaling bootserver scaling_dhcp network scaling_net leasetime 600</pre>	

Table 32 *Shutdown-timeout*

Syntax	<pre>shutdown-timeout <target> <value></pre>	
Description	Sets how long the system will wait after a shutdown has begun until a forced reboot is done.	
Options	<target>	Target node(s).
	<value>	Time in seconds.
Examples	<pre>shutdown-timeout all 360</pre>	



Table 33 SSH

Syntax	<code>ssh <target> <network></code>	
Description	Restrict SSH to listen to a specific network. This parameter can be defined multiple times if SSH should listen to more than one network. Note: if <code>ssh</code> is not defined for a blade, no restriction on SSH is made, meaning it will listen to all available interfaces.	
Options	<code><target></code>	Target blade(s).
	<code><network></code>	Name of the network that SSH should listen to.
Examples	<code>ssh payload internal</code>	
Exceptions	If the <code>ssh</code> keyword is defined for a network, SSH cannot be used towards movable IPs on that network. If SSH access to movable IPs is required, all <code>ssh</code> parameters should be removed and <code>iptables</code> used to restrict SSH traffic.	

Table 34 SSH.port

Syntax	<code>ssh.port <target> <port number></code>	
Description	The port that the SSH service should listen for connections on, instead of the default 22.	
Options	<code><target></code>	Target blade(s).
	<code><port number></code>	The network port that SSH should listen on for connections. Value range is 1 - 65535
Examples	<code>ssh.port all 1024</code>	

Table 35 SSH.rootlogin

Syntax	<code>ssh.rootlogin <target> <value></code>	
Description	Disable root login over the SSH connection. Note, if <code>ssh.rootlogin</code> is not defined for a blade, the default is to permit root login over SSH.	
Options	<code><target></code>	Target blade(s).
	<code><value></code>	Either <code>on</code> to permit root login over SSH or <code>off</code> to not permit root login over SSH.
Examples	<code>ssh.rootlogin all off</code>	

Table 36 Syslog



Table 36 Syslog

Syntax	<pre>syslog <target> <facility> <file> syslog <target> facility <facility> file <file> syslog <target> facility <facility> host <hostname:port> syslog <target> facility <facility> host <hostname:port> file<file></pre>	
Description	<p>Filters a syslog facility to a separate file or to a remote host over the network.</p> <p>This parameter allows syslog facility local0 - local7 to be filtered out from the default log file (/var/log/<hostname>/messages) and instead be stored in a separate file (/var/log/<hostname>/<file>) or dispatched over the network to a remote host <hostname:port>. Note that logs originating from payload blades are stored on both control blades (stored in /var/log/<hostname>/<file>) and payload blades (stored in /var/log/<file>). This parameter can be defined multiple times if more facilities should be filtered. However, one specific facility can only be filtered to one file.</p> <p>Note: Log rotation is not automatically performed on the new log files, neither on control blades nor payload blades. This must be taken care of by configuring appropriate mechanisms to ensures that the log file is rotated properly. Failure to rotate log files is potentially dangerous as it could fill up the file system, causing unpredictable results. Note that log rotation must be performed on log files stored on control blades as well as the log files stored on payload blades.</p>	
Options	<code><target></code>	Target blade(s).
	<code><facility></code>	Name of the facility to filter (local0 - local7).
	<code><file></code>	Name of the file where log entries should be stored. This file will be stored as /var/log/<hostname>/<file> on control blades and /var/log/<file> on payload blades.
	<code>hostname:port</code>	hostname Hostname or IP address of remote syslog machine port listening port of the remote syslog
Examples	<pre>syslog all local0 mylog0 syslog control local1 mylog1 syslog payload local2 mylog2 syslog 3 local3 mylog3 syslog control facility host 10.35.28.41:48484 local7 file mylog7</pre>	



Table 37 Timezone

Syntax	timezone <zone>	
Description	Defines the time zone of the cluster. If no time zone is defined, UTC is assumed.	
Options	<zone>	The time zone, using standard Linux time zone notation (Europe/Stockholm, GMT+1, etc).
Examples	timezone GMT+1 timezone Europe/Stockholm	

Table 38 TIPC

Syntax	tipc <target> <attribute> <attribute_value> Note: When <attribute> is an address then <target> must be an <id>.	
Description	Defines an attribute for TIPC and sets a value to this attribute.	
Options	<target>	Target node(s).

Table 38 TIPC

Attributes	attributes	attribute value	
	<address>	<interface 1> ... <interface n>	<p><address>: TIPC address (that is, Z.C.N).</p> <p><interface X>: Each interface that is to be used by the TIPC must be defined, using the interface name defined in an interface statement.</p>
	dynamic	<interface 1> ... <interface n>	<p>dynamic: the address will be automatically assigned based on to the node ID. For example, if node 2 wants an address it will be given the TIPC address 1.1.2</p> <p><interface X>: Each interface that is to be used by the TIPC must be defined, using the interface name defined in an interface statement.</p>
	link_priority	<integer>	<p>link_priority: the TIPC link priority</p> <p><integer>: Value range is 0 (lowest) - 31 (highest).</p> <p>A value of 32 (the default) tells TIPC to use the standard priority associated with the bearer's media type.</p> <p>This parameter is only available on newer versions of tipc, 2.0 or later.</p>
	link_tolerance	<ms>	<p>link_tolerance: the TIPC link tolerance</p> <p><ms>: Value range is 50ms - 30000ms.</p> <p>This parameter is only available on newer versions of tipc, 2.0 or later.</p>
	link_window	<integer>	<p>link_window: the TIPC link window</p> <p><integer>: Value range is 16 - 150.</p> <p>This parameter is only available on newer versions of tipc, 2.0 or later.</p>
	networkid	<integer>	<p>networkid: the TIPC network id number</p> <p><integer>: Value range is 1 - 9999.</p>



Table 38 TIPC

Examples	<pre>tipc all dynamic bond0 tipc 1 1.1.143 bond0 tipc all networkid 4371 tipc all link_tolerance 1500</pre>
Note:	<p>Attribute <code>max_ports</code> is not supported any more, the attribute cannot be changed and is static in the TIPC kernel module.</p>

Table 39 Watchdog

Syntax	<code>watchdog <target> soft ipmi auto</code>	
Description	<p>Defines the default or fallback watchdog to use.</p> <p>The default is to use the hardware watchdog if present (<code>auto</code>), other watchdogs are used only if no hardware watchdog is present.</p> <p>For more information about watchdog refer to the relevant section in the <i>LDE Programmer's Guide</i>, Reference [11]</p>	
	<code><target></code>	Target blade(s).
Examples	<pre>watchdog all ipmi watchdog payload soft watchdog control auto</pre>	

5.2.3 Requirements and Guidelines

The following section provides requirements and guidelines for creating a cluster configuration.

- An internal network must be defined. This network must be named `internal` and have a network prefix of 24.
- The `nfs` parameter must be defined with an IP address on the internal network.
- A `mip` with the name `nfs` must be defined. This should be given the same IP address as the `nfs` parameter (see above).
- The `boot` parameter must be defined with an IP address on the internal network.
- A `mip` with the name `boot` must be defined. This should be given the same IP address as the `boot` parameter (see above).

For a load-balancing boot setup on LDEwS, replace the `mip` named `boot` with two `mips` named `boot_a` and `boot_b`. The same rules as for `boot` apply to these `mips`.

- For every `mip` assigned to an interface, that specific interface must also be defined using the `interface` parameter. This is of course also valid for the mandatory `nfs` and `boot` `mip`. For interfaces that should have multiple `mips`, use the `alias` interface type.
- To set up a default route, a `network` with address and prefix set to `0.0.0.0/0` must be configured. A route should then be defined to use this network.

Note: Mandatory options are required on single node installations and on clustered (normal) installations, although some options are ignored on single node installations.

5.2.4 Change and Update

The cluster configuration is read by each blade during boot. When any change has been done to the cluster configuration, it should be reloaded on each blade, running `lde-config --reload --all`

In general, a blade only needs to be rebooted if the configuration affecting that particular blade is changed. However, there is one exception:

- A control blade is added to the configuration. In this case, the whole cluster needs to be rebooted.

For example, if the network address of a blade is changed, only that blade needs to be rebooted. If, on the other hand, the network address of the internal network is changed, the whole cluster must be rebooted since all blades are affected by such a change.

Adding or removing blades in the cluster is therefore possible without having to reboot other blades as such configuration changes only affect one blade. This allows expansion and repair of hardware without having to reboot the whole cluster.

5.2.4.1 Validate

When the configuration has been changed, a basic validation of the configuration can be made with the command:

```
lde-config --validate
```

5.2.4.2 Reload

When the configuration has been changed it must to be reloaded on **all** blades before rebooting to take affect. Even if the configuration changes does not



affect a particular blade, e.g. adding a new blade, it must still be reloaded on all blades to take affect. To reload the configuration issue the command:

```
lde-config --reload
```

To reload the cluster configuration on a remote blade, issue the command:

```
lde-config --reload --node <node id>
```

For example:

```
lde-config --reload --node 3
```

It is also possible to reload the cluster configuration on all blades in the cluster by issuing the command:

```
lde-config --reload --all
```

5.2.5

Example

```
# # Example /cluster/etc/cluster.conf
# node 1 control control1
node 2 control control2
node 3 payload payload1
node 4 payload payload2

network internal 192.168.0.0/24
network traffic 192.168.1.0/24
network nbi 192.168.2.0/24
network default 0.0.0.0/0
interface 1 eth0 ethernet 00:02:B3:B8:AC:01
interface 1 eth1 ethernet 00:02:B3:B8:AC:02
interface 1 eth2 ethernet 00:02:B3:B8:AC:03

interface 2 eth0 ethernet 00:02:B3:B8:AC:04
interface 2 eth1 ethernet 00:02:B3:B8:AC:05
interface 2 eth2 ethernet 00:02:B3:B8:AC:06

interface 3 eth0 ethernet 00:02:B3:B8:AC:07
interface 3 eth1 ethernet 00:02:B3:B8:AC:08
interface 3 eth2 ethernet 00:02:B3:B8:AC:09

interface 4 eth0 ethernet 00:02:B3:B8:AC:0A
interface 4 eth1 ethernet 00:02:B3:B8:AC:0B
interface 4 eth2 ethernet 00:02:B3:B8:AC:0C

interface all bond0 bonding eth0 eth1
interface all eth2.100 vlan

interface control eth2.200 vlan
```

```
interface control bond0:1 alias
interface control bond0:2 alias
interface control bond0:3 alias

interface control eth2.200:1 alias

tipc all dynamic bond0

ip all bond0 internal dynamic
ip all eth2.100 traffic dynamic

ip control eth2.200 nbi dynamic

mip control nfs bond0:1 internal 192.168.0.100
mip control boot_a bond0:2 internal 192.168.0.101
mip control boot_a bond0:3 internal 192.168.0.102
mip control oam eth2.200:1 nbi 192.168.2.100

route all default gateway 192.168.1.1

timezone Europe/Stockholm

ntp 192.168.1.120
ntp 192.168.1.121
ntp 192.168.1.122

dns all 192.168.1.130
dns all 192.168.1.131

nfs 192.168.0.100

boot 192.168.0.101
boot 192.168.0.102

ipmi all 192.168.0.200
root watchdog all soft

# End of file
```

5.3 The Etc-Overlay-Framework

The `etc-overlay-framework` is intended for use when complete files under the `/etc` directory need to be overwritten and this needs to be done prior to normal system startup.

A good example of a use of the `etc-overlay-framework` is for hardening changes where the configuration must be applied before networking is configured and the changes should be made in a non-dynamic fashion in order to prevent damaged configuration files.



Two distribution specific etc-overlays are provided by LDE, one default and one hardened. The contents of each is described in Section 5.3.3 on page 42

5.3.1 Deploying an Etc-Overlay on LDE with SLES

On LDE with SLES, the etc-overlay RPM should be installed using the cluster rpm command:

```
cluster rpm --add <etc-overlay-rpm-filename> -n <node id>
[reboot node]
```

After installation the etc-overlay RPM can NOT be activated with the "**--activate**" command, an error message will appear, if this is attempted. To fully activate the new configuration (and effectively overlay the files under `/etc` ,) the node must be rebooted. During startup, the system will print out information about the overlayed folders/files.

On a single node only one etc-overlay RPM can be installed, even if the rpm files have different names. To install a new, differently named etc-overlay RPM on a node, first the previous rpm must be fully removed.

The etc-overlay RPM on a given node can be upgraded using the "**cluster rpm --upgrade**" command, just as any other rpms (although rebooting the system is necessary in this case too). To fully remove an etc-overlay from a system, the "**cluster rpm --remove**" command must be used, then the targeted node must be rebooted.

On a disk cache enabled LDE with SLES system the rootfs will be cleaned whenever the etc-overlay is upgraded or removed during the subsequent reboot.

On a standalone system the etc-overlay is not currently enabled.

5.3.2 Deploying an Etc-Overlay on LDE for RHEL and SLES

On LDE for RHEL, the etc-overlay RPM should be installed using the normal rpm command:

```
rpm -i <etc-overlay-rpm-filename>
[reboot node]
```

Notes: To fully apply the etc-overlay configuration on the system, the targeted node needs to be rebooted. During shutdown, the `K50lde-etc-config` init script will apply the new configuration on the system.

There can be services which are not stopped yet, using configuration files overwritten by the etc-overlay. These services may be adversely affected.



If the installation was successful, the system will print out information about which folders/files were overwritten or created during each shutdown.

If applying the configuration during shutdown fails for some reason, it will be applied instead during boot and the system will reboot an extra time to enforce the configuration.

On a single node only one etc-overlay RPM can be installed, even if the rpm files have different names. To install a new, differently named etc-overlay RPM on a node, first the previous rpm must be removed using the `rpm -e` command or some other package handling command.

The etc-overlay RPM can be upgraded/removed from the system using the common `rpm` command with its built-in flags.

The removal of the etc-overlay from a node does NOT restore the original files of the system, which were affected by the previously used etc-overlay; instead, the system will simply not apply the overlay configuration on `/etc/` during the following shutdown sequences.

5.3.3 The provided etc-overlays

Two etc-overlays are provided, one default and one hardened. This chapter contains a summary of the different modifications made to the default RHEL and SLES configuration files in the two etc-overlays. Which modifications which will be present in the default and hardened overlay respectively is described in table Table 40

Table 40 Overlays Provided by LDE

Modification	default-overlay	hardened-overlay
Default Umask 027		yes
Legal Warning at Login		yes*
Inactivity Timer for Login Session		yes
Inactivity Timer for User Account		yes
Strong Password Enforcement		yes
Auditing - Full Personal Accountability		yes

5.3.3.1 Default Umask 027

The support for a default umask of 027 is provided by modifying the `/etc/login.defs` file and modifying the line containing `UMASK` :

```
UMASK                                027
```

This works for accounts that have not had a specific umask set during account creation.



On RHEL the file `/etc/login.defs` is also modified to remove `umask` setting on certain accounts.

5.3.3.2 Legal Warning at Login

The legal warning at Login is a banner shown during the login process containing legalese specific to the customer.

For local logins over serial console the banner will be shown before the user name and password prompt.

For remote logins over SSH the banner will be shown when the remote SSH client connects to the SSH server. For some SSH clients, if generating their own user name prompt, the banner may appear between the user name and password prompts.

The banner included with LDE is empty by default so an application will need to modify the file `/etc/issue` (for serial console) and `/etc/issue.net` (for remote login over SSH).

For local logins over serial console the system is pre-configured and the `etc-overlay` contains no configuration for handling this. For remote login over SSH the provided configuration changes are made to the file `/etc/sshd/sshd_config`:

```
Banner /etc/issue.net
```

Users needs to have the proper **LogLevel** configured on the client side in order to not suppress the legal warning.

5.3.3.3 Inactivity Timer for Login

Support is only implemented for SSH session timeout. It is provided by modifying the `etc/ssh/sshd_config` file and adding two lines containing:

```
ClientAliveInterval 1800
ClientAliveCountMax 0
```

If there is no activity, the session will disconnect after 30 minutes.

5.3.3.4 Inactivity timer for User Accounts

The support for inactivity timer for user accounts is provided by modifying the PAM configuration of the system. Specifically on SLES the file `/etc/pam.d/common-session-lde` is created (pointed to by the `/etc/pam.d/common-session` symlink) and on RHEL the file `/etc/pam.d/system-auth-lde` is created (pointed to by the `/etc/pam.d/system-auth` symlink) and also the file `/etc/pam.d/sshd` is modified inserting the following line in all the files:

```
session optional pam_exec.so seteuid /usr/sbin/account-expiry
```

This will execute the program `/usr/sbin/account-expiry` on each login. This program moves forward the account expiry by 90 days each time. NOTE:

Accounts which should not expire must belong to the group `no_expire`. The root user will of course never expire.

To re-enable the account, the root user has to use the `usermod` command, and set an appropriate expiry date for the user.

On LDE for the `lde-hardening` RPM is needed as this provides the `/usr/sbin/account-expiry` executable.

5.3.3.5 Strong Password Enforcement

Strong password enforcement is provided by modifying the PAM configuration. The modified files are `/etc/pam.d/common-password` on SLES and `/etc/pam.d/system-auth-lde` on RHEL (symlinked from `/etc/pam.d/system-auth`.) The following lines are added (the `\` signifies line continuation):

```
password requisite pam_cracklib.so difok=3 minclass=3 minlen=8 \
    maxrepeat=3 retry=3 reject_username
password requisite pam_pwhistory.so remember=5
use_authtok
```

The above changes enforce the following rules:

- Passwords shall be at least 8 characters in length.
- Passwords shall contain at least 3 of the following elements: at least one lower case alpha character, at least one upper case alpha character, at least one numeric character, at least one special character.
- Passwords must be no more than three of the same characters used consecutively.
- No real names, words, either in combination with numbers in front or at the tail
- Passwords must not be a repeat or the reverse of the associated user ID.
- Each new login password shall differ from the previous password. The degree of difference is at least 3 character positions
- Passwords must be no more than three of the same characters used consecutively.
- Each node support a password history to prevent password reuse. At least 5 unique new passwords have to be associated with a user account before an old password can be reused.

5.3.3.6 Auditing - Full Personal Accountability

Full personal accountability entails the ability to log watch O&M actions are taken by users logged in to the system. This is accomplished through enabling the Linux auditing framework.



The pam configuration files `/etc/pam.d/system-auth-lde` (symlinked from `/etc/pam.d/system-auth`) and `/etc/pam.d/password-auth-lde` (symlinked from `/etc/pam.d/password-auth`) on RHEL and `/etc/pam.d/common-session-lde` (symlinked from `/etc/pam.d/common-session`) on SLES are modified to add the following line:

```
session required pam_tty_audit.so enable=*
```

The common audit dispatch daemon configuration file `/etc/audit/plugins.d/syslog.conf` is modified to contain the following:

```
active = yes
direction = out
path = builtin_syslog
type = builtin
args = LOG_INFO LOG_LOCAL0
format = string
```

sending the audit logs to syslog local facility 0.

Also, on SLES, the audit daemon configuration file `/etc/audit/audit.rules` is filled with the following content:

```
# First rule - delete all
-D
```

```
# Increase the buffers to survive stress events.
# Make this bigger for busy systems
-b 320
```

```
# Feel free to add below this line. See auditctl man page
-e 1
```

which enables the audit daemon.

When logs are written to the syslog each line typically receives a header consisting of the timestamp and the issuing host/machine that sent the log. In order to read this logs with `aureport` for further investigation, one has to remove this header. Recommended way to do this is using `sed`:

```
$ sed 's/^. *audispd: //' /path/to/log_file > outputfile
$ aureport --tty -i outputfile
```

5.3.4 Modifying the provided etc-overlays

The two (per-distribution) provided etc-overlays may be modified or completely replaced. Modification is the preferred route as this allows the users to receive the latest updates as they come.

The etc-overlay consists of a directory structure containing configuration files, which will overwrite files under `/etc` during system boot. This directory structure is packaged inside an rpm, which is deployed on the system like any other RPM.

To create an etc-overlay RPM, the `lde-etc-overlay.py` tool should be used, and no other RPM creation method is supported. This tool can be used

on any Linux distribution (specifically tested on Ubuntu 12.04, SLES 11 SP2, RHEL 6.4) and should be used as part of the applications build system.

All commands of the tool have the '`-h`' (help) flag, where more information can be found about that command, and its parameters.

5.3.4.1 Step 1. Create the configuration

The configuration must be created within a directory structure, from which the etc-overlay RPM will be built. This directory structure must conform to these rules:

- Should have a 'toplevel' directory, which contains the configuration. The name of this directory will be the input for the `create-rpm` command.
- The toplevel directory should contain only nodegroup-folders. These nodegroup-folders can be grouped into 2 categories, and must conform to the following naming rules:

- The pre-defined nodegroups: `all`, `control`, `payload`
- User defined nodegroups: `<nodegroup-name>.<precedence-level>`, where the `<nodegroup-name>` must be an already defined nodegroup on the cluster (defined in the `cluster.conf` configuration file).

Precedence-levels must be unique and greater than 0. The higher the precedence-level, the later it will be applied, meaning that files contained in the folders with higher precedence level WILL overwrite files in folders with lower precedence. See the examples section Section 5.3.4.6 on page 48

- All files inside of custom nodegroup-folders have higher precedence than the built-in (`all/control/payload/`) nodegroup-folders:

Precedence-order: "all" < "control" < "payload" < custom nodegroups

NOTE: The data/folder hierarchy inside these nodegroup-folders will be applied on the system relative to the `/etc/` folder by default, so it should not contain the `etc` directory itself.

5.3.4.2 Step 2. Verify the directory structure (Optional)

The created directory structure can be verified using the `lde-etc-overlay.py` tool, to check that it conforms to the rules defined within the etc-config-framework, or not. To verify the configuration, use the following command line:

```
lde-etc-overlay.py verify -f <toplevel-folder-name>
```



NOTE: During RPM creation, the directory structure will be verified automatically, so this step is fully optional.

5.3.4.3

Step 3. Create the RPM

The etc-overlay RPM can be created with the following command line (line-breaks represented by '\')

```
lde-etc-overlay.py create-rpm <toplevel-folder-name> \
    -f <custom-name> -V <version> -R <release>
```

Naming convention of the built etc-overlay RPM file:

```
<custom-name>-etc-overlay-<version>-<release>.noarch.rpm
```

Mandatory parameters:

- The only mandatory parameter is the `<toplevel-folder-name>`, which holds the configuration data.
- The etc-overlay RPM can be built from a complying tarball too, in that case the input parameter is the name of this tarball.

Optional parameters:

- `<custom-name>`: the name of the rpm file can be customized using this parameter. When no custom-name is given, the etc-overlay RPM file will be created as:

```
etc-overlay-<version>-<release>.noarch.rpm
```

- `<release>`: defines the release of the RPM, default value: 0
- `<version>`: defines the version of the RPM, default value: 0.1

If there is some kind of verification problem with the created configuration within the toplevel directory, the `lde-etc-overlay.py` tool will not create the etc-overlay RPM, and will show an error message. Besides this message, the tool can be used to debug the problem using the verbosity (`-v`) parameter, where 1 is the lowest, and (currently) 3 is the highest value for this command.

5.3.4.4

Etc-Overlay Tarball

There is a possibility, to create a tarball from the toplevel directory, instead of an etc-overlay rpm file. In a simple etc-overlay RPM scenario, where an application wants to configure the system in their on custom way, this is not needed at all.

This feature can be useful, when e.g. multiple teams work together, and want to modify different parts of the same system. As only 1 etc-overlay RPM file



can be installed on 1 node, the teams can create their own modifications, build a tarball using the `lde-etc-overlay.py` tool, and then pass this tarball on for the next team, which can use the `lde-etc-overlay.py` tool to unpack this file, and make further modifications for the configuration. It is important to use the `lde-etc-overlay.py` tool to create/extract the mentioned tarball, and NOT use any other common command for this task.

A complying tarball can be created using the following command:

```
lde-etc-overlay.py pack <toplevel-folder-name>
```

and unpacked using the command:

```
lde-etc-overlay.py unpack <tarball-filename>
```

5.3.4.5 Notes

It is suggested to create backups of the files, which will be overlayed on the system.

Do NOT use the '**apply**' command of the `lde-etc-overlay.py` tool, as it might apply some unwanted configuration on the current machine (especially when the tool is used on one of the target nodes).

5.3.4.6 Overlay Examples

Example 1.

When a node is part of multiple nodegroups, and the nodegroup-folders have different versions of the same file, the file with the higher precedence-level will be applied to the targeted node. E.g., in case of a toplevel structure and nodegroup definitions like the following:

```
all
  file1.txt
  file2.txt
control
  file2.txt
group_a.1
  file1.txt
group_b.2
  file2.txt
group_c.3
  file1.txt
```

```
All nodes:      1 2 3 4 5
Control nodes:  1 2
Payload nodes:  3 4 5
Group A nodes:  1 3 4
Group B nodes:  2
Group C nodes:  4
Applied configuration:
```



```
Node 1:
  file1: from "group_a.1", overrides "all"
  file2: from "control", overrides "all"
Node 2:
  file1: from "all"
  file2: from "group_b.2", overrides "all" and "control"
Payload 3:
  file1: from "group_a.1", overrides "all"
  file2: from "all"
Payload 4:
  file1: from "group_c.3", overrides "all" and "group_a.1"
  file2: from "all"
Payload 5:
  file1: from "all"
  file2: from "all"
```





6 Kickstart Management (for LDEfR)

6.1 Overview

The LDE for Redhat Enterprise Linux (LDEfR) system uses the kickstart installation method to achieve an automated installation for a blade. LDEfR provides kickstart files that can be used for installing control and payload blades. The kickstart files used are stored at `/cluster/etc/templates/kickstart/`. These can be changed if that is needed for any specific application need.

6.2 Creating Kickstart structure for a blade

To be able to install a blade using kickstart, the correct file structure and configuration files need to be created. Create the structure by using the following command:

```
lde-kickstart --node <node id>
```

For example:

```
lde-kickstart --node 3
```

Once the structure is created, the blade can be installed by booting up the blade using PXE.





7 Software Configuration (for LDEfS and LDEfR)

7.1 Overview

The system uses RPM as the software package format. Each blade in the system has its own local RPM database. Packages can be added, upgraded, and removed independently on each blade. A local software repository is maintained on the shared disk, accessible from all blades.

7.2 Repository Management

The following commands allow management of the local software repositories.

7.2.1 List Repositories

It is possible to list all configured software repositories.

`lde-repo --listrepos` will give the names of all repositories configured.

During a standard LDE installation the following repositories are created:

- Distribution base, for example RHEL.Key verification
- Distribution updates, for example RHupdates
- Ericsson base
- Ericsson updates

It is recommended that any update from the distribution is stored in the distribution update repository and to store any Ericsson provided software in one of the Ericsson repositories. However, an application can use the repositories in any way they want.

7.2.2 Create Additional Repository

To create an additional repository use the following command:

```
lde-repo --create <repository name>
```

This will create an empty new yum repository as well as create a default configuration for the repository. See Section 7.2.4 Create and Update Repository Configuration on page 54 for how to set additional configuration for the repository.



For example:

```
lde-repo --create AppRepository
```

Note: The configuration needs to be created for each blade, see Section 7.2.4 Create and Update Repository Configuration on page 54 for details how to do it

7.2.3 Remove Repository

To remove an repository use the following command:

```
lde-repo --Remove <repository name> [--all|--node <nodeid>]
```

For example:

```
lde-repo --Remove AppRepository --all
```

7.2.4 Create and Update Repository Configuration

If configuration needs to be created for a new repository or it needs to be updated for existing repository, the following command should be used:

```
lde-repo ----update <repository name> [options]  
[--all|--node <nodeid>]
```

The `options` can be one of several of:

`--full-name <description>` - to give a more meaningful name to the repository

`--key <path to key file to be used>` - specify which key file should be used verify packages in this repository

`--cost <number>` - give importance to the repository. A lower number gives higher importance.

For example:

```
lde-repo --update AppRepository --full-name "Repository  
for application" --key /etc/pki/application/application-  
key --all
```

7.2.5 Add a Software Package to a Repository

To add an RPM package from a repository use the following command:

```
lde-repo --import <repository name> <package name 1> ...  
<package name n>
```

For example:



```
lde-repo --import EricssonUpdates application-R1A01-0.x
86_64.rpm
```

7.2.6 Remove a Software Package from a Repository

To remove an RPM package from a repository use the following command:

```
lde-repo --remove <repository name> <package name 1> ...
<package name n>
```

For example:

```
lde-repo --remove EricssonUpdates application-R1A01-0.x
86_64.rpm
```

7.2.7 List Software Packages in Repository

It is possible to list all RPM packages that are in a repository.

`lde-repo --list <repository name>` will list the RPMs in a specific repository.

For example:

```
lde-repo --list EricssonUpdates
```

7.3 Software Management

The following commands allow management of the RPM packages used by each blade.

Note: Whenever doing an RPM add/remove/upgrade, a local snapshot can be used for doing a rollback of the blade and its running software if the software management change failed for some reason. Creating a local snapshot is described in Section 12.2 on page 75 and restoring a local snapshot is described in Section 12.4 on page 76.

7.3.1 Adding an Software Package

To add a new RPM package to a blade, make sure the package exists in one of the repositories. Then add the software package using the following command:

```
lde-rpm --add <package name> --node <node id>
```

For example:

```
lde-rpm --add application-R1A01-0.x86_64.rpm --node 3
```



Note: When a new RPM has been added to a blade, an activation must be issued to actually install the new RPM. See Section 7.3.7 on page 57 for details.

7.3.2 Removing an Application Package

To remove an RPM package from a blade use the following command:

```
lde-rpm --remove <package name> --node <node id>
```

For example:

```
lde-rpm --remove application-R1A01-0.x86_64.rpm --node 3
```

Note: When an RPM has been removed from a blade, an activation must be issued to actually remove the RPM. See Section 7.3.7 on page 57 for details.

7.3.3 Upgrading an Application Package

To upgrade an RPM package on a blade, make sure the package exists in one of the repositories. Then update the software package using the following command:

```
lde-rpm --upgrade <package name> --node <node id>
```

For example:

```
lde-rpm --upgrade application-R1A01-0.x86_64.rpm --node 3
```

Note: When an RPM has been upgraded on a blade, an activation must be issued to actually upgrade the RPM. See Section 7.3.7 on page 57 for details.

7.3.4 Listing Packages

It is possible to list all RPMs that are currently used by a blade.

`lde-rpm --list --node <node id>` will list the RPMs used by a specific blade.

`lde-rpm --list --all` will list the RPMs used by all blades.

For example:

```
lde-rpm --list --node 3
```

7.3.5 List Uncommitted Package Changes

It is possible to list all package changes that have not been committed with an `lde-rpm --activate` for a specific blade.



`lde-rpm -L --node <node id>` will list uncommitted changes on a specific blade.

For example:

```
lde-rpm -L --node 3
```

7.3.6 Drop Uncommitted Package Changes

It is possible to drop all package changes that have not been committed with an `lde-rpm --activate` for a specific blade.

`lde-rpm -D --node <node id>` will drop uncommitted changes on a specific blade.

For example:

```
lde-rpm -D --node 3
```

7.3.7 Activating Packages

After packages have been added, removed, or upgraded, it is possible to commit the changes on the affected blade. By activating the configuration, affected packages will be installed, removed and upgraded all depending on the changes made. The following command should be used to activate RPMs on a blade.

```
lde-rpm --activate --node <node id>
```

For example:

```
lde-rpm --activate --node 3
```

7.3.8 Check for Upgraded Packages

It is possible to check if any packages used by the blade should be upgraded based on if any newer version of the RPMs have been added to the repositories.

```
lde-rpm --checkupgrade --node <node id>
```

For example:

```
lde-rpm --checkupgrade --node 3
```

7.3.9 Automatically Upgrade of Packages

It is possible to automatically upgrade all packages used by a blade to latest version in the repository. In this case, no `lde-rpm --activate` is needed.



```
lde-rpm --autoupgrade --node <node id>
```

For example:

```
lde-rpm --autoupgrade --node 3
```

7.4 Upgrading To New System Version

This section provides a step-by-step instruction of how to do a system version upgrade. This could be a LDE upgrade, a RHEL upgrade or a combination of both.

1. In case a new LDE release is to be upgraded, import LDE RPMs to wanted repository. The repository would typically be `Ericsson` or `EricssonUpdates`, but it is the choice of the application

```
lde-repo --import <repository name> path_to_lde/*
```

2. If a new release or update of RHEL is included in the upgrade, it needs to be copied to the local repository. If there is no new RHEL release/update, continue at step Step 5.

Mount RHEL ISO file or DVD drive

To mount iso file: `mount -o loop <iso> <RHEL ISO mount point>`

To mount DVD drive `mount /dev/sr0 <RHEL ISO mount point>`

3. Replace local RHEL repository

```
lde-repo --distro <RHEL ISO mount point> --progress
```

4. If any RHEL updates exist based on the old base RHEL repository, these can safely be removed. To recreate an empty `RHupdate` repository do the following steps:

```
lde-repo --Remove RHupdates --all
```

```
lde-repo --create RHupdates
```

```
lde-repo --update RHupdates --all. See Section 7.2.4 Create and Update Repository Configuration on page 54 for details what configuration can be set for the repository.
```

5. Start upgrading the first control blade.
6. Whenever an extensive set of RPM should be upgraded it is preferred to shutdown the `lde-failoverd` service on the control blade that should be upgrade to avoid disturbance during upgrade.

```
/etc/init.d/lde-failoverd stop
```



7. If a kernel update is included in the upgrade set it is recommended to start with this package (to find out the upgrade set run `lde-rpm --checkupgrade --node <node id>`).

```
lde-rpm --upgrade kernel -n <node id>
```

8. If any kernel module is included in the upgrade set or should be added to the system, these should be included now.

Upgrade example: `lde-rpm --upgrade tipc-km-2.6.32_220.el6.x86_64 -n <node id>`

Add example: `lde-rpm --add tipc-km-2.6.32_220.el6.x86_64 -n <node id>`

9. Activate the new upgraded software

```
lde-rpm --activate -n <node id>
```

10. Upgrade the rest of the upgrade set

```
lde-rpm --node <node id> --autoupgrade
```

11. Complete the upgrade by rebooting the blade. This is normally only needed if the kernel or any kernel module have been upgraded but still preferred to make sure that the system is fully restarted with the new system version.

```
lde-reboot --node <node id>
```

12. If the kernel was upgraded and the old kernel is not needed anymore, it can be removed at this point.

```
lde-rpm --remove kernel-<kernel-version> --node <node id>
```

```
lde-rpm --activate --node <node id>
```

Example of kernel version remove: `lde-rpm --remove kernel-2.6.32-71.el6.x86_64 --node <node id>`

13. Do the upgrade for the second control blade, starting at Step 6.
14. Do the upgrade for each payload blade in the system, starting at Step 7.





8 Software Configuration (for LDEwS)

The system uses the Redhat Package Manager (RPM) as the software package format. Each node in the cluster has its own software configuration, which is simply a list of RPMs that are to be installed on the node at every startup. Package can be added, upgraded, and removed independently on each node.

Since the root file system on all nodes is RAM based the RPMs will automatically be installed on the nodes every time they boot up. If the persistent root file system is used, then the root file system is not always erased and recreated on every startup, but only when an Operating System (OS) RPM is upgraded.

It is possible to force a reinitialization of the root file system by using command `cluster rootfs -c -o -n <node_id>` before a reboot.

The root file system will be erased and recreated on the next reboot after an OS RPM installation. It is not possible to disable this behavior.

Note: The order among packages when installing/removing/upgrading packages is determined by the internal RPM dependencies among the packages.

When using any Service Availability Framework (SAF) distribution, software packages are handled as described in the documentation of the respective SAF distribution.

8.1 Adding an Application Package

To add a new RPM package to a node, perform the following steps:

1. Copy the new RPM package to the RPM repository located at `/cluster/rpms`.
2. Update the software configuration:

```
cluster rpm --add <package name> --node <node id>
```

```
Example: cluster rpm --add application-R1A01-0.i586.rpm
--node 3
```

Note: When a new RPM has been added to a node, an activation or a node reboot must be issued to actually install the new RPM. For more information about activating packages, see Section 8.7 Activating Packages on page 65.

8.2 Removing an Application Package

To remove an RPM package from a node, perform the following step:



1. `cluster rpm --remove <package name> --node <node id>`

Example: `cluster rpm --remove application-R1A01-0.i586.rpm --node 3`

Note: When an RPM has been removed from a node, an activation or a node reboot must be issued to actually remove the RPM. For more information about activating packages, see Section 8.7 Activating Packages on page 65.

8.3 Upgrading an Application Package

To upgrade an RPM package on a node, perform the following steps:

1. Copy the new RPM package to the RPM repository located at `/cluster/rpms`
2. Update the software configuration:

`cluster rpm --upgrade <package name> --node <node id>`

Example: `cluster rpm --upgrade application-R1A01-0.i586.rpm --node 3`

Note: Whenever an RPM upgrade is done, a snapshot can be used to do a rollback of the RPM configuration. Creating snapshots are described in Section 12.3 Creating a Local Snapshot on LDEwS on page 75 and restoring snapshots are described in Section 12.5 Restoring a Local Snapshot on LDEwS on page 76. A snapshot includes all node configurations, such as RPMs on other nodes and the cluster configuration file.

When an RPM has been upgraded on a node, an activation or a node reboot must be issued to actually upgrade the RPM. For more information about activating packages, see Section 8.7 Activating Packages on page 65.

8.4 Upgrading an LDEwS Package

Note: Whenever an LDEwS OS RPM upgrade is done, a snapshot can be used to do a rollback of the RPM configuration. Creating snapshots are described in Section 12.3 Creating a Local Snapshot on LDEwS on page 75 and restoring snapshots are described in Section 12.5 Restoring a Local Snapshot on LDEwS on page 76. A snapshot includes all node configurations, such as RPMs on other nodes and the cluster configuration file.

To upgrade an OS RPM on a node, perform the following steps:

1. Obtain the RPM from the runtime deliverable (see Section 8.4.1 Extracting LDEwS Packages from Runtime Deliverable on page 63) and copy it to the RPM repository located at `/cluster/rpms/`.



2. Update the software configuration:

```
cluster rpm --upgrade <package name> --node <node id>
```

```
Example: cluster rpm --upgrade ldews-payload-cxp9020125
-4.0.0-1.sle12.x86_64.rpm --node 3
```

To activate the LDEwS software, a reboot must be issued on the node. To avoid downtime, a rolling upgrade approach may be used where one node of the cluster is rebooted at the time. Ensure that a rebooted control node is fully booted (all software running) before rebooting the other control node. This can be accomplished by waiting for the disk synchronization and only reboot the other control node when the login prompt is shown on the newly rebooted control node.

Note: RPMs for control nodes must be upgraded locally on the control nodes for the node to be able to boot from disk using the new version. If this is not done, the RPM will warn the user by issuing the message `NOTE: RPM installed offline`. The RPM can be committed to the boot disk of the control node by issuing a package synchronization, see Section 8.6 Synchronizing Packages on page 65.

8.4.1 Extracting LDEwS Packages from Runtime Deliverable

The LDEwS OS RPMs are distributed in Software Delivery Package (SDP) files inside the runtime deliverable. The SDP files are GZIP compressed GNU tar files, which means that they can be extracted the same way as a `.tar.gz` file or by the `sdp2rpm.sh` script included in the runtime deliverable.

Example

```
node1:/tmp # tar zxvf ldews-4.0.0-runtime-sle-cxp9020125.
tar.gz
control
ERIC-LINUX_CONTROL-CXP9013151_4.sdp
ERIC-LINUX_PAYLOAD-CXP9013152_4.sdp
IDENTITY
install
install_tspasf_packages
payload
sdp2rpm.sh
node1:/tmp # ./sdp2rpm.sh ERIC-LINUX_CONTROL-CXP9013151_4.
sdp
node1:/tmp # ./sdp2rpm.sh ERIC-LINUX_PAYLOAD-CXP9013152_4.
sdp
node1:/tmp # ls *.rpm
ldews-control-cxp9020125-4.0.0-1.sle12.x86_64.rpm
ldews-payload-cxp9020125-4.0.0-1.sle12.x86_64.rpm
node1:/tmp #
```



The RPMs may now be copied to the `/cluster/rpms/` directory and then upgraded.

8.4.2 Upgrading LDEwS and Other RPMs

This is an implementation of an LDEwS upgrade (during which other RPMs may be upgraded also). This path will provide a no downtime RPM upgrade with support for rollback of the cluster configuration including the RPMs using a snapshot.

To upgrade LDEwS and other RPMs, perform the following steps:

1. (Optional) Create a snapshot. For more information, see Section 12.3 Creating a Local Snapshot on LDEwS on page 75.
2. Update the RPMs of LDEwS and other applications.
3. Perform a rolling reboot of the cluster nodes as follows:
 - a Reboot one of the control nodes, wait for it to be available for operation (that is, wait until the login prompt is shown on the console or until the Distributed Replicated Block Device (DRBD) is fully synchronized). If possible and applicable, perform checks to verify that all software is up and running on the node.
 - b Reboot the other control node. Wait for it to be rebooted and available for operation (see a above) if the cluster is large (more than 10 nodes) to make sure that the load balancing is in place before continuing.
 - c Reboot all payload nodes, one node at the time. Verify after each reboot that the upgrade is successful and that the applications on the node work properly.
4. Ensure that the cluster is working as intended.

The upgrade is completed.

If a failure is detected during the upgrade, make a restore of the snapshot and a rolling reboot according to roll back to the cluster (and software) configuration in place before the upgrade started. For more information about restoring snapshots, see Section 12.5 Restoring a Local Snapshot on LDEwS on page 76.

8.5 Listing Packages

To list all RPMs that are currently used by a node, perform the following step:

1. `cluster rpm --list --node <node id>`

Example: `cluster rpm --list --node 3`



To list all RPMs used by all nodes, perform the following step:

1. `cluster rpm --list --all`

8.6 Synchronizing Packages

Synchronization of RPM packages is typically useful after restoring the system from a snapshot or a backup. A synchronization will essentially reinstall the OS RPM specified in the software configuration for a node. It will also verify that all packages used by the node are available in the repository.

To synchronize the RPMs on a node, perform the following step:

1. `cluster rpm --sync --node <node id>`

Example: `cluster rpm --sync --node 3`

8.7 Activating Packages

After packages have been added, removed, or upgraded, it is possible to make the change on the affected node without rebooting it. By activating the configuration, affected packages will be installed, removed, and upgraded, all depending on the changes made. If the OS RPM of the node has been upgraded, the activation will fail and a reboot is required to get the node to run the correct software.

To activate the RPMs on a node, perform the following step:

1. `cluster rpm --activate --node <node id>`

Example: `cluster rpm --activate --node 3`

Note: If an activation is not issued, the changes done in the RPM configuration of the node will take affect until the node is rebooted.





9 User and Group Management

9.1 Overview

Users and groups can be either local or global. Local users and groups only exist on the blade where they are created. Global users and groups are on the other hand cluster wide and exist on all blades in the cluster, not just the blade on which the user or group was created.

9.2 Adding/Removing/Modifying Users and Groups

Adding, removing and modifying users and groups is done using standard Linux tools (`useradd`, `userdel`, `groupadd` and `groupdel`).

9.3 Make a User or Group Global

Users and groups are by default local to the blade there they were created. To make a created user or group global use the following command:

```
lde-global-user --user <user name>
```

For example:

```
lde-global-user --user admin
```

To make a group global:

```
lde-global-user --group <group name>
```

For example:

```
lde-global-user --group adminGroup
```

When a global user or a global group is removed locally from one blade, the user is also removed globally.

Note: The user or group that should be global must exist locally on the blade where the command is issued. Also make sure that all groups that a global user belongs to are global or are available on blades in the system.

9.4 Changing Password

Changing password for users (both local and global) is done using standard Linux tool `passwd`.



Note: Changing the password of the `root` user will not affect the password that has to be used when the blade is booted up in maintenance mode.

9.5 Account and Password Aging

When a user account is created, no account or password aging are applied by default. However, by using standard Linux tool, `chage`, it is possible to set up both account and password aging. It is for example possible to set up the maximum days a password is valid, how many days before expiration a warning message should be displayed to the user at log in, and for how many days an account can have an expired password before the account is inactivated. The table below gives some hits of good practice values when defining account and password aging. The values should be considered as recommendations and can of course be changed at any time when suitable.

Table 41 Account and password aging

Parameter	Value
account expire date	-1 (disabled)
account inactive date	30
minimum number of days before password can be changed	0
maximum number of day before password must be changed	90
number of days prior to password expiring a warning message should be give to user (at login)	7

To apply the above values for a user, the following command should be issued:

```
chage <user name> --mindays 0 --maxdays 90 --expiredate -1  
--inactive 30 --warndays 7
```

The `chage` (1) man page gives more details about syntax and available options.

9.6 Forced Password Change

When a user account is created or reactivated, it might be given a default password that should be changed the first time the user logs into the system. To force the user to change the password, the following command should be issued after the user account has been created or reactivated:

```
passwd -e <username>
```

The `passwd` (1) man page gives more details of available options.



9.7 Allowing Login Users (for LDEwS)

External connectivity to the cluster is provided through the SSH and a serial cable. By default, only `root` is allowed to login externally to the nodes in the cluster.

To make a newly created user able to login externally, its user name and to which nodes it is allowed to login to must be added to the `/cluster/etc/login.allow` file. The following syntax must be used:

```
<user> <target>
```

The syntax of `<target>` is the same as for the cluster configuration. For more information, see Section 5.2.1 Legend on page 9. This configuration file is valid for all nodes in the cluster.

For example, if `root` is to be allowed to login to control nodes and user `tspsaf` is to be allowed to login to all payload nodes, `/cluster/etc/login.allow` must look as follows:

```
root control tspsaf payload
```

Ensure that the `/cluster/etc/login.allow` file is owned by `root` and that the file has the access permission `640` (that is owner read/write, group read, other no access). Incorrect access permission will cause the file to be ignored.

Note: No restrictions are applied once a user is logged in on a node and wants to log in to another node in the cluster.

9.8 Login Information

By default, no information is provided to the user when logging in using a CLI based shell. However, the system provides the file `/cluster/etc/motd`, which allows a text message to be displayed to the user every time it logs in to the system. The same message is displayed to all users and on all blades in the cluster.

9.9 Inactivity Logout

If a login session should be logged out due to inactivity, a shell profile (i.e. `/etc/profile.d/`) should be added to the system that sets up the correct shell variable with the desired time out value. To set this for a bash shell, which is the default shell in the system, the **TMOUT** variable should be set to the number of seconds the session can be idle until an auto logout is made. For a tcsh shell, the **autologout** variable should be set to the number of minutes the session can be idle before auto logout.





10 Logging In Remotely

10.1 SSH

The system provides SSH for remote login. To login on one of the blades in the cluster, issue the following command:

```
ssh -l <user> <blade>
```

Where *<user>* is a user available on the system and *<blade>* is either the host name or the IP address of the blade.





11 Backup

11.1 Overview

Creating a complete system backup involves taking a copy of the shared replicated file system (`/cluster/`) and storing that data in an archive file on a remote server. The archive will be of the format TAR, compressed with GZIP. The remote server must be accessible using SSH. The system can later be restored to the state it had when the backup was taken. The backup does not contain any files from the root file system (apart from the shared replicated file system files).

11.2 Creating a Backup

Follow the steps below to create a backup and store it on a remote server.

1. Login as `root` on one of the control blades.
2. Start taking a backup by issuing the following command:

```
lde-backup --create <user>@<server>:</path/file.tar.gz>
```

Where `<user>` is a user name available on the remote server, `<server>` is the host name or IP address of the remote server and `</path/file.tar.gz>` is the location and file name where the backup will be stored on the remote server.

Note: This operation may load the cluster file system so that other applications have trouble accessing it. This can be relieved by adding the `--safe` flag, that will reschedule the backup process in a lower scheduling class. This will make the process only use resources when available, which in turn will make the process slower.

3. Enter the password for `<user>` on `<server>` when prompted.
4. The system will now start to create the backup. This can take a while depending on how much data is stored on the file system and the connection speed to the remote server. If the backup completed successfully the following text will be displayed:

```
Backup completed
```

11.3 Restoring a Backup

Please contact Ericsson personnel for restoring a backup.





12 Local Snapshot

12.1 Overview

Creating a local snapshot will create a snapshot of the root file system of a blade. This should typically be used as a pre step before any changes to the RPMs used by the blade should be applied. If an upgrade or any change to the running RPMs on the blade would fail, restore of an local snapshot will bring back the system to the state of when the local snapshot was created. A blade can only manage one local snapshot at a time.

Note: The difference to a full backup is that local snapshot will not make a complete backup of the whole shared replicated disk, but instead only include the local root file system of the blade.

12.2 Creating a Local Snapshot on LDEfS

Use the command below to create a local snapshot.

```
lde-local-snapshot --create --node <node id>
```

For example:

```
lde-local-snapshot --create --node 3
```

The local snapshot is successfully created when the text 'Snapshot completed' is displayed.

12.3 Creating a Local Snapshot on LDEwS

To create a snapshot, perform the following steps:

1. Login as `root` on one of the control nodes.
2. Start taking a snapshot:

```
cluster snapshot --create </path/file.tar.gz>
```

`</path/file.tar.gz>` is the location and file name where the snapshot will be stored.

Note: This operation may load the cluster file system so that other applications have trouble accessing it. This can be relieved by adding the `--safe` flag, which will reschedule the backup process in a lower scheduling class. The process will then only use resources when available, which in turn will make the process slower.



3. The system starts to create the snapshot. If the snapshot completes successfully the text `Snapshot completed` is shown.

12.4 Restoring a Local Snapshot on LDEfS

Use the command below to restore a local snapshot.

```
lde-local-snapshot --restore --node <node id>
```

For example:

```
lde-local-snapshot --restore --node 3
```

The local snapshot is successfully restored when the text 'Snapshot restore done' is displayed. However, the blade needs to be rebooted to complete the restore.

Note: The local root file system will not be restored until the blade has been rebooted.

12.5 Restoring a Local Snapshot on LDEwS

To restore a snapshot, perform the following steps:

1. Login as `root` on one of the control nodes.
2. Start the restore of the snapshot:

```
cluster snapshot --restore </path/file.tar.gz>
```

`</path/file.tar.gz>` is the location and file name where the snapshot is stored.

Note: This operation may load the cluster file system so that other applications have trouble accessing it. This can be relieved by adding the `--safe` flag, which will reschedule the backup process in a lower scheduling class. The process will then only use resources when available, which in turn will make the process slower.

3. The system starts to restore the snapshot. If the restore completes successfully the text `Snapshot restore completed` is displayed.
4. After the restore is completed, a package synchronization must be made for each node in the cluster. Issue the following command for each node:

```
cluster rpm --sync --node <node id>
```

For more information about synchronizing packages, see Section 8.6 Synchronizing Packages on page 65.



12.6 Delete a Local Snapshot on LDEfS

If the created local snapshot is not needed anymore, typically when the upgrade was successful and there is no need be able to rollback to earlier state, it can be deleted using the command below.

```
lde-local-snapshot --delete --node <node id>
```

For example:

```
lde-local-snapshot --delete --node 3
```

A local snapshot can be deleted from the blade that owns the snapshot or from a control blade.

12.7 BRF (for LDEwS)

LDEwS participates in a BRF backup as a Persistent Storage Owner. The operation of such backup is controlled via BRF-C. For disaster recovery and extra-ordinary situations the low-level command line interface described here can be used.

This command will back up data in the persistent storage areas (see Persistent Storage API). An important detail here is that the persistent data types 'Configuration data' and 'Software data' are refer to as 'System data', i.e. an backup containing 'System data' will contain both configuration and software.

Operations available through the `lde-brf` (the same command is also available as `cluster brf`) command are:

- Create backup
- Restore backup
- Delete backup
- Export backup
- Import backup
- Various list and informational functions

Create, restore and delete operations are performed in two steps:

- Prepare for operation
- Commit/Cancel operation

A backup that is prepared and committed for restore will be restored at the next reboot of the cluster, it is important to perform a **full** single step cluster reboot when activating a backup.

12.7.1 Backup creation

12.7.1.1 Prepare creation of backup

```
lde-brf create -l <label-name> -t system|user -m <metadata>
```



Prepare creation of a backup of type system or user data or prepare creation of a backup with a combination of the two backup types.

12.7.1.2 Commit/Cancel the creation of a backup

When a backup creation has been prepared it can either be committed or canceled.

```
lde-brf create --commit -l <label-name>
```

```
lde-brf create --cancel -l <label-name>
```

Note: You can cancel a committed backup (this is actually a deletion of the backup without any prepare/commit stages!)

12.7.1.3 Example

Note: It is normally **not** recommended to manually create backups since the backups will be unusable by BRF-C if the meta-data will not match a BRF-C initiated backup

```
lde-brf create -l example1 -t system -m 'Backup example, system'
```

Prepares creation of a backup with system (config and software) data

```
lde-brf create --commit -l example1
```

Commits the backup 'example1' prepared above

12.7.2 Backup restore

12.7.2.1 Prepare restore of a backup

```
lde-brf restore -l <label-name> -t system|user
```

Prepares a backup for restore

12.7.2.2 Commit/Cancel the restore of a backup

```
lde-brf restore --commit -l <label-name> [-u]
```

```
lde-brf restore --cancel -l <label-name>
```

Note: You can cancel restore of a committed backup as well as cancel restore of a backup prepared, but not committed, for restore.

12.7.2.3 Example, system data restore backup

```
lde-brf restore -l example2 -t system
```

```
lde-brf restore --commit -l example2
```



Commits the restore of the backup 'example2', the system state backed up in this backup will be restored at next cluster reboot.

12.7.2.4 Example, user data backup

The user backup type is the only type that can be restored without a reboot if you supply the '-u' option e.g

```
lde-brf restore -l example2 -t system
lde-brf restore --commit -l example2 -u
```

Restores the backup immediately without a reboot provided it is a user backup that is being committed.

Note: The option to activate the data without a reboot (-u) is optional and without that the user data will be activated at next cluster reboot. This option is ignored if there is no user data prepared for restore

12.7.3 Backup delete

12.7.3.1 Prepare deletion of a backup

```
lde-brf delete -l <label-name> -t system|user
```

12.7.3.2 Commit/Cancel deletion of a backup

```
lde-brf delete --commit -l <label-name>
```

```
lde-brf delete --cancel -l <label-name>
```

12.7.3.3 Example

```
lde-brf delete -l example1 -t system
```

Prepares the backup 'example1' created above for deletion

```
lde-brf delete --commit -l example1
```

Commits the deletion of backup 'example1' prepared for deletion above

12.7.4 Print functions

```
lde-brf print <print type> [-t system|user]
```

Prints information about backups

<print type> is one of:

metadata, prints the descriptions of every backup

labels, prints labels of available backups

latest, prints the label of latest created backup

restored, prints the label of latest restored backup

state, prints the internal state



reboot, action taken at next reboot

12.7.4.1 Examples

12.7.4.1.1 Print type metadata

```
lde-brf print metadata -t system
```

```
example1 metadata  
example2 metadata
```

Note: Meta-data is used by BRF-C to store special formatted information about a backup and should not be considered to be human-readable information.

12.7.4.1.2 Print type labels

```
lde-brf print labels -t user
```

```
user: example1  
user: example2
```

12.7.4.1.3 Print type latest

```
lde-brf print latest
```

```
config: example2  
software: example2  
user: example2
```

```
lde-brf print latest -t system
```

```
config: example2  
software: example2
```

12.7.4.1.4 Print type restored

Prints information about the label of the latest restored backup

```
lde-brf print restored  
user: example1
```

Note: the printout is empty if no backup was restored at the last reboot of the cluster

12.7.4.1.5 Print type state

```
lde-brf print state
```

```
idle
```



Provides information if a operation is running (output will then be working)

12.7.4.1.6 Print state reboot

```
lde-brf print reboot
```

example1 containing 'system' data will be activated during next reboot

12.7.5 Export and import of backups

```
lde-brf export -l <label> -t system|user -f <file/path>
```

Creates a file containing a backup. If the argument to **-f** is a path (i.e. /tmp/) and not a file the filename will automatically be set to <label>.tar

```
lde-brf import -l <label> -t system|user -f <file>
```

Creates a backup from a file containing a backup (previously exported).

12.7.6 View active backup labels

Active backup labels are automatically created during the BRF backup operations if both **lde-brf-script** and BRF 1.2 or later are installed on the system.

Active backup labels can only be viewed by **lde-brf**.

Table 42 lde-brf activelabel

Syntax	<code>lde-brf activelabel --print <active_backup_label> -t <type></code>
Description	Command for viewing active backup labels



Options	<code><active_backup_label></code>	Backup label type. Can be one of the following: <ul style="list-style-type: none">• <code>all</code> Print all active backup labels of the given backup type• <code>lastcreated</code> Print the last created backup of the given backup type• <code>lastimported</code> Print the last imported backup of the given backup type• <code>lastexported</code> Print the last exported backup of the given backup type• <code>lastrestored</code> Print the last restored backup of the given backup type• <code>prc</code> Print the primary restore candidate of the given backup type
	<code><type></code>	Backup type. Can be one of the following: <ul style="list-style-type: none">• <code>user</code>• <code>system</code>
Examples	<pre>lde-brf activelabel --print prc -t user lde-brf activelabel --print lastcreated -t system</pre>	

12.7.7

Examples

Create a backup, for a system using BRF-C the backup should be created through the functionality offered by BRF-C.

1. Prepare creation of a backup of all types:
`lde-brf create -l example -t system -m 'An example'`
2. Print created labels:



- ```
lde-brf print labels
config: example (incomplete)
software: example (incomplete)
```
3. Commit the creation  
**lde-brf create --commit -l example**
  4. Print created labels:  
**lde-brf print labels**  
config: example  
software: example
  5. Print latest created label:  
**lde-brf print latest**  
config: example  
software: example
  6. Print metadata:  
**lde-brf print metadata**  
An example
  7. Export the backup to file /cluster/example:  
**lde-brf export -l example -f /cluster/**

#### Import and restore backup

1. Import a backup  
**lde-brf import -l example2 -f /cluster/example2.tar**
2. Print labels:  
**lde-brf print labels**  
config: example  
config: example2  
software: example  
software: example2
3. Prepare for restore of the backup example2:  
**lde-brf restore -l example2 -t system**
4. Commit the restore:  
**lde-brf restore --commit -l example2**
5. Print the label to be restored at next reboot:  
**lde-brf print reboot**  
example2 containing 'software' 'config' data will be  
activated during next reboot
6. Reboot the cluster:  
**cluster reboot -q --all**
7. When cluster has rebooted, print restored backup:  
**lde-brf print restored**  
example2





## 13 Reboot

### 13.1 Overview

The system provides means to remotely reboot any blade in the system. Note that the blade that is to be rebooted must be up and running for this to work. Also note that **only** control blades are allowed to reboot other blades. You must therefore be logged in on a control blade when issuing the commands below.

### 13.2 Reboot a Single blade

To reboot a single blade in the system, issue the following command on a control blade.

```
lde-reboot --node <node id> Or lde-reboot --hostname
<hostname>
```

For example:

```
lde-reboot --node 3
```

```
lde-reboot --hostname b11
```

### 13.3 Reboot the Whole Cluster

To reboot all blades in the system, issue the following command on a control blade.

```
lde-reboot --all
```

**Note:** This command will first issue the reboot command consecutively to all of the payload nodes, and only once they have all ceased running will the control nodes be rebooted. If, for any reason, the other control node is rebooted before the last of the payload nodes, then some payload nodes may start up again. When this occurs, the control node issuing the original reboot will not recognize that all the payload nodes have ceased running, and will be caught in a deadlock. Ensure no other reboot commands are issued on the other control node once a cluster reboot has been initiated.

### 13.4 Bypass the BIOS When Rebooting

By passing the parameter `--quick` to the cluster reboot command, it will re-execute the kernel that would be executed during the next reboot without



doing a hardware reset. On hardware that has long power on self-test (POST) cycles, this would shorten the time required for a system or single blade reboot.

Example, reboot a single blade:

```
lde-reboot --quick --node <node id>
```

Example, reboot all blades in the system:

```
lde-reboot --quick --all
```

**Note:** The success of the BIOS bypassing reboot depends much on the blade hardware and the hardware drivers. Some drivers might expect the hardware to be in a certain state when loaded, which can cause trouble and hangs. Make sure that the reboot mode is tested on the hardware before usage in the field.



## 14 Power

### 14.1 Overview

The system provides means to remotely power on, off and reset of blades as well as querying the power status. Note that remote power management requires the system to support IPMI over LAN.

### 14.2 Configuration

To be able to use power management over IPMI, it has to be configured in the cluster configuration. See Section 5.2.2 Parameters on page 10 for how to configure IPMI.

When issuing any of the power related commands, a prompt will appear asking for the IPMI password configured for the IPMI user.

### 14.3 Power On

To power on a blade use the following command.

```
lde-power --on --node <node id>
```

For example:

```
lde-power --on --node 3
```

### 14.4 Power Off

To power off a blade use the following command.

```
lde-power --off --node <node id>
```

For example:

```
lde-power --off --node 3
```

### 14.5 Power Reset

To reset a blade use the following command.

```
lde-power --reset --node <node id>
```



For example:

```
lde-power --reset --node 3
```

## 14.6 Power Status

To check the power status on a blade use the following command.

```
lde-power --status --node <node id>
```

For example:

```
lde-power --status --node 3
```



## 15 Alarms

### 15.1 Overview

The system provides a tool for monitoring the status of the alarms that can be raised. For instructions and details about raised alarms, see the separate OPI documents Reference [1], Reference [2], Reference [3], Reference [4], Reference [5] and Reference [6]

### 15.2 Alarm Status

To get the current status of the alarms, issue the following command.

```
lde-alarm --status --node <node id>
```

For example:

```
lde-alarm --status --node 3
```

It is also possible to get the status from all blades by issuing the following command.

```
lde-alarm --status --all
```





## 16 Configuration Management (for LDEwS)

Some of the services in the base operating system are configured by LDEwS to provide functionality such as cluster internal configuration, shared file system and network boot of nodes. The changes can be categorized into static changes and dynamic changes. Static changes are done to the default files when the LDEwS image is created and dynamic changes are applied to the system in runtime and are related to the cluster configuration.

It is possible to apply additional changes to these services. If the change is to one of the statically changed files then the change can be performed by installing an RPM on the system which will perform the change to the service or file in question. If the change is to one of the files that are updated in runtime it is not enough to update the file during startup since it might be overwritten during runtime. To modify such a file LDEwS provides an interface into the runtime configurational. This interface allows a user to plug in a configuration script that is executed along with the standard LDEwS configurations. This means that the change will be reapplied every time the configuration is reloaded in runtime.

*Table 43 Statically and dynamically changed configuration files*

| Files that are changed statically       | Files that are changed dynamically |
|-----------------------------------------|------------------------------------|
| /var/lib/nfs/state                      | /etc/drbd.d/global_common.conf     |
| /etc/modprobe.d/blacklist               | /etc/drbd.d/drbd0.res              |
| /etc/bindresvport.blacklist             | /etc/dhcpd.conf                    |
| /lib/udev/rules.d/50-udev-default.rules | /etc/exports                       |
| /etc/bash.bashrc                        | /etc/HOSTNAME                      |
| /etc/logrotate.d/ntp                    | /etc/hosts                         |
| /etc/logrotate.d/rsync                  | /etc/security/limits.conf          |
| /etc/logrotate.d/syslog                 | /etc/ssh/sshd_config               |
| /etc/logrotate.d/syslog-ng              | /etc/ssh/sshd_config_internal      |
| /etc/logrotate.d/wtmp                   | /etc/ssh/ssh_config                |
| /etc/logrotate.d/xinetd                 | /etc/logrot.d/default              |
| /etc/xinetd.conf                        | /etc/mactab                        |
| /etc/pam.d/sshd                         | /etc/ntp.conf                      |
| /etc/pam.d/login                        | /etc/resolv.conf                   |
| /var/lib/dhcp/db/dhcpd.leases           | /root/.shosts                      |
| /etc/init.d/boot.udev                   | /etc/ssh/ssh_known_hosts           |



|                                        |                               |
|----------------------------------------|-------------------------------|
| /etc/ld.so.conf.d/perl-workaround.conf | /etc/syslog-ng/syslog-ng.conf |
|                                        | /etc/localtime                |

Configuration scripts should be installed into `/usr/lib/lde/config-management/` and install symlinks pointing to that file into any of the following locations:

- `/usr/lib/lde/config-management/start`
- `/usr/lib/lde/config-management/stop`
- `/usr/lib/lde/config-management/config`

Links should be created in start and stop if the script will need to restart some service. The order of a configuration reload is: stop, config & start. The links in start, stop and config should be prefixed by S<number>, K<number> and C<number> respectively. Scripts will be executed in this order and numbers over 500 should be used by applications.

The scripts will be called with:

- param 1: action - start, stop or config
- param 2: generate phase - init or reload
- param 3: root path where config should be generated

After the config management scripts have been installed a configuration reload should be run:

```
cluster config -r
```

Note: Any change to files or services needed by LDEwS could potentially disable LDEwS functionality and should only be done if the change required is to application specific and thus not possible to get into the product itself. If a such a file is modified it is very important to check if the changed file is changed when taking in a new version of LDEwS.

## 16.1 Example

This example describes building an RPM to update the node SSHD configuration.

### 16.1.1 File layout

The RPM (mynode-ssh-config-R1A01.noarch.rpm) installs the following files:

```
/usr/lib/lde/config-management/mynode-sshd
/usr/lib/lde/config-management/start/S500mynode-sshd -> ../mynode-sshd
```



```
/usr/lib/lde/config-management/stop/K500mynode-sshd -> ../mynode-sshd
/usr/lib/lde/config-management/config/C500mynode-sshd -> ../mynode-sshd
```

### 16.1.2 Sample script

```
#!/bin/bash

update_config() {
 <make changes to/etc/sshd_config here>
}

ACTION="$1"
PHASE="$2"
ETC_ROOT=${3:-"/etc"}

case "$ACTION" in
 start)
 /etc/init.d/sshd start
 ;;
 stop)
 /etc/init.d/sshd stop
 ;;
 config)
 update_config
 ;;
 *)
 echo "Usage $0 start|stop|config <phase> <etc root>" >&2
 exit 2
 ;;
esac
```

### 16.1.3 Sample RPM spec file

```
#
mynode-ssh-config.spec: RPM build specification
#

Name: mynode-ssh-config
Version: R1A
Release: 0
License: Proprietary
Group: Applications
Vendor: Ericsson AB
Summary: SSH client configuration
BuildArch: noarch
BuildRoot: %(mktemp -ud %({_tmppath})/%{name}-%{version}-%{release})-X
Prefix: /usr/lib/lde/config-management
```



```
%description
Updates the ssh configuration

%install
rm -rf $RPM_BUILD_ROOT
install -m 755 -d $RPM_BUILD_ROOT/%{prefix}/start
install -m 755 -d $RPM_BUILD_ROOT/%{prefix}/stop
install -m 755 -d $RPM_BUILD_ROOT/%{prefix}/config
install -m 755 %{_sourcedir}/mynode-sshd $RPM_BUILD_ROOT/%{prefix}
ln -s %{prefix}/mynode-sshd $RPM_BUILD_ROOT/%{prefix}/start/S500mynod
ln -s %{prefix}/mynode-sshd $RPM_BUILD_ROOT/%{prefix}/stop/K500mynode
ln -s %{prefix}/mynode-sshd $RPM_BUILD_ROOT/%{prefix}/config/C500myno

%clean
rm -rf $RPM_BUILD_ROOT

%files
%defattr(-,root,root)
%{prefix}/mynode-sshd
%{prefix}/start/S500mynode-sshd
%{prefix}/stop/K500mynode-sshd
%{prefix}/config/C500mynode-sshd

End of file
```

### 16.1.4 Building the RPM

Instructions for building the RPM can be found in the documentation specific to the host Linux distribution.



## 17 Cluster tool

The overall command `cluster` is still available for backward compatibility. To find out what sub-commands are available run:

**`cluster`**

The available sub-commands will vary depending on what LDE RPMs are installed on the system.

The option for each sub-command is still the same, independent of how it is invoked. For example, `lde-power` can also be invoked using `cluster power`.





## 18 Network Services

This chapter provides information about network services in the system that are available from external networks. This information is of interest when configuring a firewall.

### 18.1 Listening Services

The listening services are shown in Table 44.

*Table 44 Listening Services*

| Port(s)   | Protocol(s) | Description                                             | Listening Only on the Internal Network                                                                |
|-----------|-------------|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| 22        | TCP         | Secure shell (SSH/Secure File Transfer Protocol (SFTP)) | No, configurable                                                                                      |
| 67        | UDP         | Dynamic host configuration service (DHCP)               | Yes                                                                                                   |
| 69        | UDP         | File transfer service (TFTP)                            | Yes                                                                                                   |
| 111       | TCP/UDP     | Portmap service (portmap)                               | No. The service is listening on the external network, but it is serving only on the internal network. |
| 123       | UDP         | Time synchronization service (NTP)                      | No                                                                                                    |
| 514       | UDP         | Log service (syslog)                                    | Yes                                                                                                   |
| 1022      | TCP         | SSH/SFTP (only for LDEwS)                               | Yes                                                                                                   |
| 1128      | TCP         | Alarm service                                           | Yes                                                                                                   |
| 1129–1131 | TCP/UDP     | Node service                                            | Yes                                                                                                   |
| 2049      | TCP/UDP     | Network File System (NFS)                               | No. The service is listening on the external network, but it is serving only on the internal network. |
| 7788      | TCP         | Disk replication service (DRBD)                         | Yes                                                                                                   |



It is not possible to mount the NFS share over any network apart from the internal. If you want to block these services on the external network you need to apply the following iptables rules.

```
iptables -A INPUT -p <protocol> --match multiport
--destination-port <ports_to_be_blocked> -j REJECT
--reject-with icmp-port-unreachable -s <external-network>
```

Where <protocol> is {tcp, udp}, <ports\_to\_be\_blocked> are 22, 11, 2049 and <external-network> is any external network that you have configured on the blade. You need to repeat the command for each protocol and each network.

The recommended way to configure iptables is via the cluster.conf. See Table 15 for more information. For example:

```
iptables control -A INPUT -p tcp --match multiport
--destination-port 22,111,2049 -j REJECT --reject-with
icmp-port-unreachable -d 10.0.0.0/24
```

```
iptables control -A INPUT -p udp --match multiport
--destination-port 22,111,2049 -j REJECT --reject-with
icmp-port-unreachable -d 10.0.0.0/24
```

If you have multiple external networks, you need to repeat the iptables rule for each network.

## 18.2 List Services

It is possible to get a list of the listening services on a running blade by issuing the following command:

```
netstat -tulp
```

## 18.3 SSH server

The standard SSH server is started up by LDE with a default configuration, listening on default port 22 on all networks. This ssh server can be configured by the user in a persistent way, see Section 16 on page 91

Additionally on LDEwS, a second SSH server is listening on port 1022 on the internal network. This is started up by LDE with a default configuration.



# 19 Preventive Maintenance

## 19.1 System Check

This section is intended to give a short summary of procedures that a system administrator should perform in order to monitor and maintain an Linux Distribution Extension installation. This section is written primarily for system administrators, and operation and maintenance technicians who are responsible for ensuring that the system runs smoothly after installation.

A technician or system administrator performing system checks needs to be familiar with the following:

- Linux
- Internet Protocol (IP) based networks
- Have access to all relevant information about the system such as IP addresses and login credentials
- Familiarity with the Linux Distribution Extension system documentation (or at least have access to the document set for future reference)

To check if there are any pending alarms in the system, issue the following command:

```
lde-alarm --status --all
```

The available alarms are associated with critical components in a Linux Distribution Extension based installation. If any alarms have been raised, refer to the alarm operating instructions Reference [1], Reference [2], Reference [3], Reference [4], Reference [5] and Reference [6] for appropriate action.

To monitor CPU and memory usage for application and system processes, standard tools such as `ps` can be used.

To display the processes sorted by CPU usage, issue:

```
ps aux --sort pcpu
```

To display processes sorted by memory usage, issue:

```
ps aux --sort rss
```

**Note:** CPU and memory usage are highly dependent on the characteristics of the applications running on the system, but generally it should not be at or close to maximum capacity.



To check that the time synchronization through NTP is working correctly, use the `ntpq` utility to (as an example) view any peers currently known to the system:

```
ntpq -c peers
```

On a control blade, one or more external peers should be configured and in use. There should be one peer marked with an asterisk and one or more marked with a plus sign. On a payload blade, one control blade should be marked with an asterisk and the other one with a plus sign.

## 19.2 Logs

The system produces logs that contain messages from the kernel, daemons, applications, etc. These logs are stored on both control blades. Each blade in the cluster produces three separate log files, these are:

|                                                 |                                                         |
|-------------------------------------------------|---------------------------------------------------------|
| <code>/var/log/&lt;hostname&gt;/kernel</code>   | Kernel messages                                         |
| <code>/var/log/&lt;hostname&gt;/auth</code>     | Messages related to users and authentication            |
| <code>/var/log/&lt;hostname&gt;/messages</code> | All messages (except those found in <code>auth</code> ) |

Note that the kernel will also print messages (kernel crash information, etc.) to the VGA or serial console. This is useful if, for example, the kernel crashes so badly that it can no longer send the log messages to the control blades. In those cases, the VGA or serial console might be the only way to catch these messages.

**Note:** Log replication between cluster blades use the standard syslog protocol. This works in a best-effort manner using UDP and may therefore drop network-transferred log entries during periods where heavy logging occurs. It is also required for a blade to be running in order for it to be able to collect log lines, meaning that a control blade will not gather logging information during downtime or during a reboot. If log data is missing for debugging purposes, make sure that the other control blade is checked as well, if the log for the control blade indicates that a reboot or other downtime has taken place during the period of interest.

## 19.3 Copying large files to shared replicated storage

It's recommended that large files that are copied to the shared storage is done as described in this section. This means copying files with a size of 10MB or more over the network and NFS to the shared replicated storage provided by the controllers. It is possible to use for example `rsync` to limit the rate in which data is sent over the network, and can be used to make sure that one single blade is not using all of the available bandwidth (network and disk). If possible, try to minimize the number of concurrent transfers of large files



within the system. This is to prevent any critical processes from being blocked for too long waiting for I/O requests to complete, which could have negative consequences for the stability of the cluster.

As an example, limiting the bandwidth usage to about 4MB per second using `rsync`:

```
rsync --archive --bwlimit=4096 <src> <dest>
```

## 19.4 Maintenance Mode

All blades can be booted up in maintenance mode.

This is done by selecting `Maintenance mode` in the GRUB boot menu. To see the GRUB boot menu you need to have a VGA or serial control attached to the machine and press any key when the following text appears early in the boot phase.

Press any key to continue.

Note that this text is only displayed for a few seconds before the machine boots using the default `Operational mode`.

Enter the `root` password when prompted.

## 19.5 Trouble Reporting

When reporting a problem in a trouble report, it is vital that logs are attached. Include the following files in the trouble report.

|                                                         |                                                                         |
|---------------------------------------------------------|-------------------------------------------------------------------------|
| <code>/cluster/etc/cluster.conf</code>                  | Cluster configuration                                                   |
| <code>/cluster/node/&lt;node id&gt;/etc/rpm.conf</code> | Software configuration for all nodes (for LDEwS)                        |
| <code>/var/log/&lt;hostname&gt;/kernel</code>           | Kernel messages from all blades                                         |
| <code>/var/log/&lt;hostname&gt;/auth</code>             | Messages related to users and authentication from all blades            |
| <code>/var/log/&lt;hostname&gt;/messages</code>         | All messages (except those found in <code>auth</code> ) from all blades |

Also, include the output from the following command executed on **both** control blades.

|                                              |                         |
|----------------------------------------------|-------------------------|
| <code>lde-alarm --status --full --all</code> | Alarm status            |
| <code>cat /proc/drbd</code>                  | Disk replication status |



|                                         |                                                                                                |
|-----------------------------------------|------------------------------------------------------------------------------------------------|
| <code>df -h</code>                      | Disk usage status                                                                              |
| <code>ip addr show</code>               | Network interface status                                                                       |
| <code>smartctl -a &lt;device&gt;</code> | SMART status (where <i>&lt;device&gt;</i> is the device name of the local disk, e.g. /dev/sda) |

Include the following out from each blade in the system.

|                       |                                         |
|-----------------------|-----------------------------------------|
| <code>lde-info</code> | Product version                         |
| <code>rpm -qa</code>  | List of all RPMs installed on the blade |

If the kernel crashes badly, it might not be possible to log in to the blade or receive messages from it. There might however be messages available on the VGA or serial console. In that case, include these messages in the trouble report.



## Reference List

- [1] *LDE, Time Synchronization*, 1/1543-CAA 901 2978/1
- [2] *LDE, Ethernet Bonding* , 2/1543-CAA 901 2978/1
- [3] *LDE, Disk Usage*, 3/1543-CAA 901 2978/1
- [4] *LDE, Disk Replication Communication*, 4/1543-CAA 901 2978/1
- [5] *LDE, Memory Usage*, 5/1543-CAA 901 2978/1
- [6] *LDE, Disk Replication Consistency*, 6/1543-CAA 901 2978/1
- [7] *LDE Glossary of Terms and Acronyms*  
*TERMINOLOGY*, 1/0033-APR 901 0551/4
- [8] *LDE Trademark Information*  
*LIST*, 1/006 51-APR 901 0551/4
- [9] *LDE Alarm List*  
*LIST*, 2/006 51-CAA 901 2978/1
- [10] *Typographic Conventions*  
*DESCRIPTION*, 1/1551-FCK 101 05
- [11] *LDE Programmer's Guide*, 1/198 17-CAA 901 2978/4