



## Red Hat Reference Architecture Series

# Deploying Highly Available SAP Servers using Red Hat Clustering

Martin Tegtmeier, Realtech

Alfredo Moralejo, Senior Consultant (EMEA GPS)

John Herr, Senior Software Engineer (Solutions Architecture)

Frank Danapfel, Software Engineer (on-site at SAP)

Version 2.0

August 2011





1801 Varsity Drive™  
Raleigh NC 27606-2072 USA  
Phone: +1 919 754 3700  
Phone: 888 733 4281  
Fax: +1 919 754 3701  
PO Box 13588  
Research Triangle Park NC 27709 USA

Linux is a registered trademark of Linus Torvalds. Red Hat, Red Hat Enterprise Linux and the Red Hat "Shadowman" logo are registered trademarks of Red Hat, Inc. in the United States and other countries.

AMD is a trademark of Advanced Micro Devices, Inc.

SAP and SAP NetWeaver are registered trademarks of SAP AG in Germany and several other countries.

ABAP is a trademark of SAP AG in Germany and several other countries.

UNIX is a registered trademark of The Open Group.

Intel, the Intel logo and Xeon are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

POSIX is a registered trademark of IEEE.

Oracle is a registered trademark of Oracle Corporation.

IBM is a registered trademark of International Business Machines in many countries worldwide.

VMware, ESX, ESXi, and vSphere, are registered trademarks of VMware, Inc.

All other trademarks referenced herein are the property of their respective owners.

© 2011 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

The information contained herein is subject to change without notice. Red Hat, Inc. shall not be liable for technical or editorial errors or omissions contained herein.

Distribution of modified versions of this document is prohibited without the explicit permission of Red Hat Inc.

Distribution of this work or derivative of this work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from Red Hat Inc.

The GPG fingerprint of the [security@redhat.com](mailto:security@redhat.com) key is:  
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Send feedback to [refarch-feedback@redhat.com](mailto:refarch-feedback@redhat.com)



# Table of Contents

1 Executive Summary.....	1
2 Introduction.....	2
2.1 Audience.....	2
2.2 Acronyms.....	3
3 Technology Overview.....	4
3.1 Red Hat Cluster.....	4
3.2 SAP components overview.....	4
3.3 Implementing SAP services in Red Hat Clustering.....	7
4 Requirements.....	8
4.1 Server Hardware.....	8
4.2 Network.....	8
5 Environment.....	9
6 Red Hat Cluster Basics.....	10
6.1 OpenAIS.....	10
6.2 CMAN.....	10
6.3 Cluster Resource Manager.....	10
6.4 Quorum.....	11
6.4.1 Qdisk.....	11
6.4.2 Additional heuristics.....	12
6.5 Fencing.....	13
6.5.1 Power Fencing Systems.....	13
6.5.2 SAN Switch Based Fencing.....	13
6.5.3 SCSI Fencing .....	13
6.5.4 Virtual Machine Fencing .....	14
6.6 Storage Protection.....	14
6.6.1 HA-LVM, CLVM.....	14
6.6.2 GFS.....	14
6.6.3 Storage Mirroring.....	15



6.7 OS Dependencies.....	15
6.8 Stretch Clusters.....	16
6.8.1 Network infrastructure requirements.....	17
6.8.2 Storage requirements.....	17
6.8.3 Quorum in stretch clusters.....	17
6.8.4 Data replication with LVM.....	17
6.8.5 Stretch cluster limitations.....	18
6.8.6 Site failure management.....	18
6.8.7 Site recovery management.....	18
6.8.8 Stretch clusters architecture review.....	19
7 Operating System Installation.....	20
7.1 OS Installation overview.....	20
7.1.1 Core Cluster Software Installation.....	20
7.2 OS Customizations.....	21
7.2.1 SAP specific OS customization.....	21
7.2.2 NTP.....	21
7.2.3 ACPI.....	21
7.2.4 Firewall.....	22
7.3 Network Configuration.....	22
7.3.1 Public/Private Networks.....	22
7.3.2 Bonding.....	22
7.3.3 Hosts file.....	23
7.4 Storage Configuration.....	23
7.4.1 Multipathing.....	23
7.4.2 Device Mapper Multipath.....	23
7.4.3 LVM.....	24
7.4.3.1 LVM Configuration.....	25
7.4.3.2 Volume Configuration.....	25
7.5 Cluster Core Configuration.....	26
7.5.1 CMAN / OpenAIS.....	27
7.5.2 Qdisk.....	28
7.5.3 Fencing.....	29
7.5.4 Cluster Nodes.....	30
8 SAP Installation.....	31
8.1 SAP Architecture.....	31
8.2 Virtual IP Addresses.....	31



8.3 File Systems.....	31
8.3.1 Local File Systems.....	32
8.3.2 Shared Storage File Systems.....	32
8.3.3 NFS Mounted File Systems.....	32
8.4 Before Starting the SAP Installation.....	32
8.5 Installation with sapinst.....	33
8.6 Installation Post-Processing.....	33
8.6.1 Users, Groups and Home Directories.....	33
8.6.2 Synchronizing Files and Directories.....	33
8.6.3 SAP Release-specific Post-processing.....	34
8.6.4 Before Starting the Cluster.....	34
8.7 Enqueue Replication Server.....	34
9 Cluster Configuration.....	35
9.1 Cluster Resources.....	35
9.2 Basic rgmanager Configuration.....	36
9.3 Failover Domains.....	37
9.4 Cluster Resources and Services.....	39
9.4.1 SAP Resources.....	40
9.4.1.1 IP.....	40
9.4.1.2 LVM.....	40
9.4.1.3 FS.....	41
9.4.1.4 SAPInstance.....	41
9.4.1.5 SAPDatabase.....	45
9.5 Dependencies.....	47
9.5.1 Resource Dependencies.....	47
9.5.2 Service Dependencies.....	47
9.5.2.1 Hard and Soft Dependencies.....	48
9.5.2.2 Follow Service Dependency.....	48
10 Cluster Management.....	49
10.1 CMAN.....	49
10.1.1 cman_tool status.....	49
10.1.2 cman_tool nodes.....	49
10.1.3 cman_tool services.....	50
10.2 rgmanager.....	50
10.2.1 clustat.....	50



10.2.2 clusvcadm.....	51
10.2.3 rg_test.....	52
11 Closing thoughts.....	53
Appendix A: Cluster Configuration Files.....	54
Appendix B: Reference Documentation.....	58
Appendix C: Revision History.....	60



# 1 Executive Summary

This paper details the deployment of a highly available SAP landscape on Red Hat Enterprise Linux (RHEL) 5.6 with the RHEL High Availability Add-On. After an introduction to the basic concepts and system requirements, this document provides detailed information about the RHEL HA Add-On, SAP NetWeaver HA installations, and cluster configuration options.



## 2 Introduction

A cluster is essentially a group of two or more computers working together which, from an end user's perspective, appear as one server. Clustering can be used to enable storage clustering, balance load among cluster members, parallel processing, and high-availability (HA). The “highly available” aspect of any cluster service indicates that it is configured in a manner such that the failure of any one cluster member, or a subsystem failure within a member, does not prevent the continued availability of the service itself.

Ensuring the highest possible availability of SAP systems is essential for success. The availability of an SAP application typically depends on an SAP application server which in turn relies on optimal availability of an underlying database. This layered software stack sits on top of an even more complex hardware infrastructure. In order to increase the availability of SAP software, redundant hardware and additional cluster software is required.

To achieve highest availability, every single-point-of-failure has to be eliminated which creates the requirement for every hardware component to be redundant. This includes networks, network-interface-cards, routers and network-switches over storage arrays, storage networks and storage interfaces to servers, power supplies, power circuits, air conditioning and whole datacenters depending on the desired level of availability.

The cluster software monitors the status of all managed services and initiates a failover to redundant server infrastructure if problems are detected. The RHEL HA Add-On provides the features necessary to make critical SAP services on RHEL highly available. This document illustrates the recommended highly available RHEL infrastructure for SAP.

When configuring HA for an SAP environment, all the software and hardware layers must be taken into consideration. Red Hat and REALTECH, with guidance from SAP, have conducted development work in order to provide a reference architecture for High Availability for SAP NetWeaver using the RHEL HA Add-On. The resulting implementation is compliant with SAP recommendations. Together with the resource agent scripts in Red Hat Enterprise Linux 5.6, this reference architecture can serve as a guide to deploying highly available SAP applications (the majority of the SAP product portfolio including ECC, SCM, SRM, etc.) based upon SAP NetWeaver technology. It is provided as is without support or liability statements. Experts at the required technologies may follow this guide when implementing highly available, Red Hat Enterprise Linux based SAP systems.

Note that cluster software has full control of all services including the starting and stopping of SAP software. Numerous customer cases have proven how a poorly configured HA environment can inadvertently decrease the availability of critical SAP systems. As such, consulting services familiar with both SAP and RHEL HA Add-On would be a cost effective investment.

### 2.1 Audience

This document addresses SAP certified technical consultants for SAP NetWeaver with experience in HA systems. Access to SAP and Red Hat information resources such as SAP Service Marketplace and the Red Hat Knowledge Base is mandatory.





## 2.2 Acronyms

Common acronyms referenced within this document are listed below.

<b>AAS</b>	SAP Additional Application Server
<b>ADA</b>	SAP Database Type MaxDB
<b>API</b>	Application Programming Interface
<b>ASCS</b>	SAP ABAP Central Services Instance
<b>CLVM</b>	Cluster Logical Volume Manager
<b>CMAN</b>	Cluster Manager
<b>DB6</b>	SAP Database Type DB2 on Linux
<b>DLM</b>	Distributed Lock Manager
<b>ERS</b>	SAP Enqueue Replication Server
<b>GFS</b>	Global File System
<b>HA</b>	High-Availability
<b>IP</b>	Internet Protocol
<b>LVM</b>	Logical Volume Manager
<b>NAS</b>	Network Attached Storage
<b>NFS</b>	Network File Server
<b>NIC</b>	Network Interface Card
<b>NTP</b>	Network Time Protocol
<b>NW640</b>	SAP NetWeaver 2004 (kernel 6.40)
<b>NW70</b>	SAP NetWeaver 7.0
<b>OCF</b>	Open Cluster Framework
<b>ORA</b>	SAP Database Type Oracle
<b>OS</b>	Operating System
<b>PAS</b>	SAP Primary Application Server
<b>POSIX</b>	Portable Operating System Interface
<b>QDISK</b>	Quorum Disk
<b>QDISKD</b>	Quorum Disk Daemon
<b>RHEL</b>	Red Hat Enterprise Linux
<b>RIND</b>	Rind Is Not Dependencies
<b>SAN</b>	Storage Area Network
<b>SCS</b>	SAP Central Services Instance (for Java)
<b>SPOF</b>	Single Point Of Failure
<b>SSI</b>	Single System Image
<b>VFS</b>	Virtual File System



## 3 Technology Overview

### 3.1 Red Hat Cluster

For applications that require maximum system uptime, a Red Hat Enterprise Linux (RHEL) cluster with RHEL HA Add-On is the solution. The RHEL HA Add-On provides two distinct types of clustering:

- Application/Service Failover - Create n-node server clusters for failover of key applications and services
- IP Load Balancing - Load balance incoming IP network requests across a farm of servers

With RHEL HA Add-On, applications can be deployed in HA configurations so that they are always operational, bringing "scale-out" capabilities to Red Hat Enterprise Linux deployments. RHEL with the RHEL HA Add-On provides a complete, ready-to-use failover solution for SAP NetWeaver.

### 3.2 SAP components overview

In an SAP NetWeaver environment, these services must be considered:

- Database
- SAP Central Services Instance (SAP enqueue and message server)
- SAP System Mount Directory (/sapmnt/<SID>)
- SAP Instances / Application Servers

According with SAP architecture and capabilities, high availability for each component included in a SAP system can be achieved by different strategies. Some components, considered *SINGLE POINTS OF FAILURE* for the whole system require a infrastructure cluster. Availability for other components can be provided by using several active instances.



The following table shows the main SAP components for ABAP systems and how high availability may be applied.

Component	Number of components	High Availability
DBMS	1 for SAP System	Infrastructure Cluster
Enqueue Server	1 for SAP System (included in ASCS instance). Enqueue Replication Server provides further enqueue server resilience.	Infrastructure Cluster
Message Server	1 for SAP System (included in ASCS instance)	Infrastructure Cluster
Dialog work process	1 or more for ABAP instance	Infrastructure Cluster or Several Active ABAP Instances
Update work process	1 or more for ABAP instance	Infrastructure Cluster or Several Active ABAP Instances
Batch work process	0 or more for ABAP instance	Infrastructure Cluster or Several Active ABAP Instances
Spool work process	1 or more for ABAP instance	Infrastructure Cluster or Several Active ABAP Instances
Gateway	1 for ABAP instance	Infrastructure Cluster or Several Active ABAP Instances
SAP System Mount Directory	1 for SAP System	Infrastructure Cluster (Highly Available NFS Service)
ICM	1 for ABAP instance	Infrastructure Cluster or Several Active ABAP Instances
Web Dispatcher	1 or several Web Dispatcher processes	Infrastructure Cluster or Several Active Instances with load balancing

**Table 3.2.1: System Configuration**



The following table shows the main SAP components for JAVA systems and how high availability may be applied.

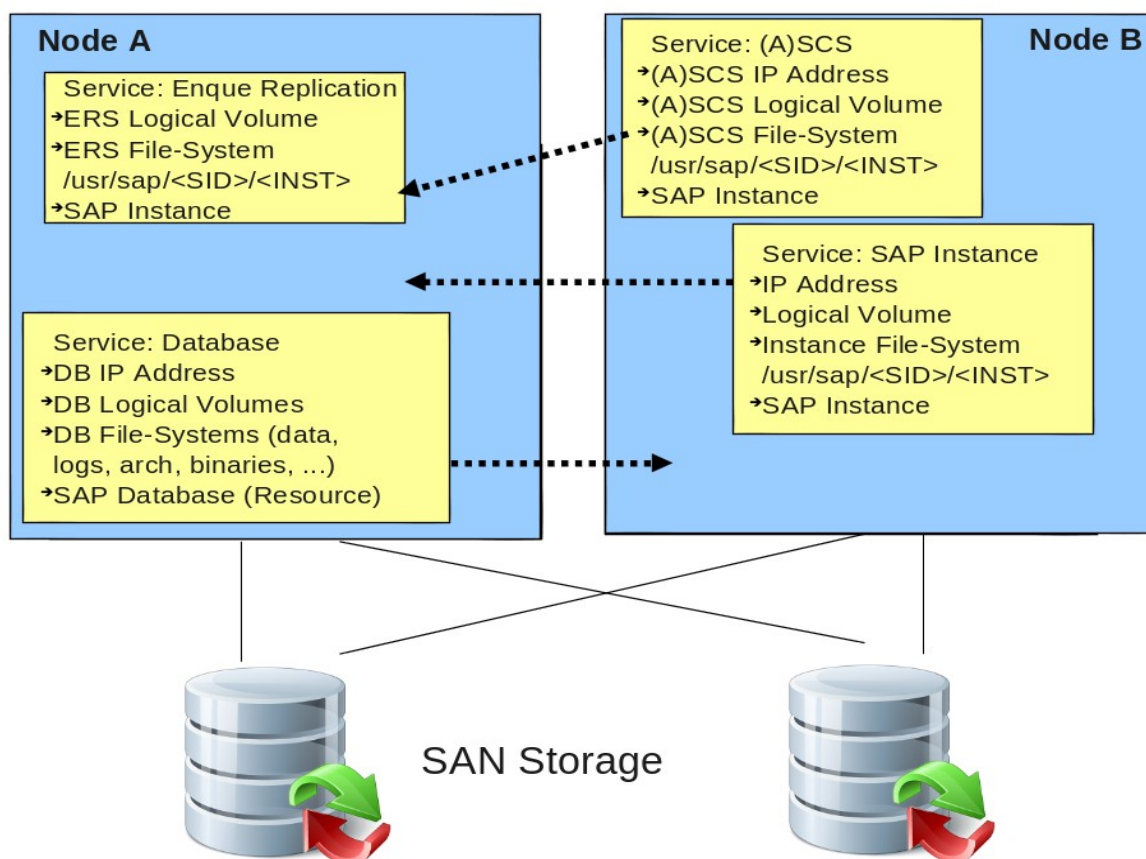
Component	Number of components	High Availability
DBMS	1 for SAP System	Infrastructure Cluster
Enqueue Server	1 for SAP System (included in SCS instance). Enqueue Replication Server provides further enqueue server resilience.	Infrastructure Cluster
Message Server	1 for SAP System (included in SCS instance)	Infrastructure Cluster
Java Dispatcher	1 for Java instance	Infrastructure Cluster or Several Active Java Instances
Java Server Process	1 for Java instance	Infrastructure Cluster or Several Active Java Instances
SAP System Mount Directory	1 for SAP System	Infrastructure Cluster (Highly Available NFS Service)
ICM (NW 7.1)	1 for Java instance	Infrastructure Cluster or Several Active Java Instances

***Table 3.2.2: Critical components in Java stack***



### 3.3 Implementing SAP services in Red Hat Clustering

The following figure shows details about the implementation of SAP components in a single stack mutual failover cluster in a two node cluster including DBMS, Central Services, Enqueue Replication Server and Application instance.



**Figure 3.3.1: SAP Cluster**

Although they are usually not single-points-of-failure, the Enqueue Replication Servers (ERS) and the SAP Application Instance(s) are controlled by the cluster software. To create a working Enqueue Replication, it is important that the ERS does not run on the same cluster node as the (A)SCS. This is because the original enqueue table lies within the same shared memory segment as the replicated table.

When the (A)SCS fails and a failover is triggered by the cluster software, the new (A)SCS must start on the node where the ERS is running. When (A)SCS comes up, it shuts down the ERS instance without cleaning its shared memory and attaches itself to the shared memory segment where the ERS had stored the replicated enqueue table. Now the replicated enqueue table has become the new original enqueue table.

To ensure that the (A)SCS "follows" the ERS instance, the follow-service dependency was implemented in the RHEL HA Add-On.

The SAP System Mount Directory should be exported by a highly available NFS server.



## 4 Requirements

### 4.1 Server Hardware

The server hardware requirements can be fulfilled by almost any enterprise grade server. The supported architectures are single or multi-core x86\_64 processors. Typically, SAP servers are equipped with a fair amount of memory starting at 8 gigabytes and are usually limited only by the hardware specification. The cluster nodes need to be attached to a fencing mechanism. Please refer to the **Fencing** section later in this document for further information.

### 4.2 Network

There should be at least two Network Interface Cards (NIC), whether embedded or added to each server. If more network interfaces are available, NIC bonding can be implemented for additional availability and is currently the only method providing NIC failover ability. One bonded network device is configured with an external IP address while the other is configured as an interconnect between cluster members using a separate network.

Clusters are highly dependent on constant communication between nodes which are maintained across the local interconnect. It is highly recommended that a dedicated non-routed private network should be used for all intra-cluster communication.

Red Hat recommends using IP multicast for cluster infrastructure traffic in Red Hat Enterprise Linux 5. Some network switches may require special configuration settings to enable multicast operation. Please refer to the hardware vendor's configuration guide for correct multicast configurations. Since version 5.6, broadcast is fully supported as an alternative to multicast in situations where multicast can not be implemented.



## 5 Environment

This section provides information about the hardware and software used to build the highly available SAP system. This information is included in the following table.

System	Specifications		
Cluster Servers [IBM i-series x86_64]	Operating System	Red Hat Enterprise Linux 5.6	
		Latest updates via Red Hat Network (RHN) subscription to channel(s)	Red Hat Enterprise Linux
			RHEL Clustering
			RHEL for SAP Applications
	Cluster Software	RHEL HA Add-On for RHEL 5.6	
		Latest updates via RHN subscriptions to channel(s):	RHEL Clustering
		Resolutions to the following Bugzilla issues	637154 - <a href="https://bugzilla.redhat.com/show_bug.cgi?id=637154">https://bugzilla.redhat.com/show_bug.cgi?id=637154</a>
			677430 - <a href="https://bugzilla.redhat.com/show_bug.cgi?id=677430">https://bugzilla.redhat.com/show_bug.cgi?id=677430</a>
	SAP Installation	SAP NetWeaver 7.0 EhP 1	
		Oracle DB 11.2.0.2	
	Storage	QLogic HBAs	
		Dell/EMC FibreChannel SAN storage array	

**Table 5.1: System Configuration**



## 6 Red Hat Cluster Basics

### 6.1 OpenAIS

In Red Hat Enterprise Linux 5.6, the core cluster infrastructure is based on the OpenAIS framework. OpenAIS is an open source implementation of the Application Interface specification defined by the Service Availability Forum, based upon extended virtual synchrony messaging. The project currently implements Application Programming Interfaces (API) application defined checkpointing, application eventing, extended virtual synchrony messaging, and cluster membership.

The heart of OpenAIS is the **aisexec** daemon, into which various services are loaded. OpenAIS can use multicast or broadcast traffic for cluster communication.

### 6.2 CMAN

Cluster Manager (CMAN) is a Red Hat specific service module that loads in the OpenAIS daemon. It provides a user API that is used by Red Hat layered cluster components. CMAN also provides additional functionality such as APIs for a quorum disk, the quorum itself, conditional shutdown, and barriers.

### 6.3 Cluster Resource Manager

The Cluster Resource Manager (**rgmanager**) manages and provides failover capabilities for cluster resource groups. It controls the handling of user requests including service start, restart, disable, and relocate.

The service manager daemon also handles restarting and relocating services in the event of failures. **rgmanager** uses Open Cluster Framework (OCF) compliant resource agents to control and monitor required resources. *SAPInstance* and *SAPDatabase* are OCF compliant resource agents provided by Red Hat.

In Red Hat Enterprise Linux 5.6, **rgmanager** includes an event driven scripting mechanism called RIND (Rind Is Not Dependencies). RIND can be used to create complex event driven service dependencies. For automatic enqueue replication failover scenarios, the RIND based *follow\_service* dependency is required.





## 6.4 Quorum

A cluster typically uses shared data resources such as a cluster file system (e.g., GFS) or local file systems controlled by the cluster resource management system (e.g., rgmanager). The cluster must be aware of the current state of the shared resources at all times. Therefore, it must be guaranteed that every critical transition within the cluster cannot compromise data integrity.

In the event of a major network problem, cluster partitioning (aka: split-brain situation) can occur. Each partition can no longer communicate with nodes outside its own partition. A Red Hat cluster requires the quorum requirement be fulfilled before a status change in the cluster is allowed. For example, quorum is required by the resource management system to relocate cluster resources or for the CMAN module to remove nodes from the cluster. The cluster partition is considered quorate if more than half of all votes within the entire cluster belong to the cluster partition. Quorum in CMAN can be defined using the following formula:

$$Q = V/2 + 1$$

where  $Q$  is the required number of votes for quorum and  $V$  is the total number of votes within the cluster.

Although the quorum requirements calculation based on the active nodes in a cluster work well for various cluster configurations, specific cases exist where the cluster cannot decide or incorrect decisions have been made.

To avoid such situations the use of a quorum disk (qdisk) has been reintroduced.

### 6.4.1 Qdisk

In specific cases, the quorum requirement based on the number of active nodes belonging to a cluster is insufficient. In a two node cluster, the standard quorum calculation ( $Q = V/2 + 1$ ) would result in two, considering one vote per cluster node. In the case of a highly available cluster, this would make no sense since it requires both nodes to be online. Therefore, the two node cluster is considered a special case and by using the `<two_node>` configuration option, quorum can be reduced to one. In this manner, quorum is maintained even if one node fails. The remaining node hosts all services managed by the cluster.

One major concern with this solution is in the case of a network loss between nodes, each node interprets the lack of connectivity as a failure of the other node. This problem is most commonly referred as a “split brain” situation, as each node is quorate by itself and assumes it is the survivor. To keep both nodes from simultaneously accessing shared resources that must not be active on more than one node at any time, each node attempts to fence the other node to prevent uncontrolled access to the shared resources. In this instance, whichever node successfully fences the other first becomes the surviving member.

A quorum disk (qdisk) can be used to prevent this situation, bolstering the quorum by adding an additional vote or votes to the cluster.

In a two node cluster configuration with a qdisk, the total expected votes would be three with a quorum of two.



In small multi-node cluster configurations, other types of problems can occur. In a three or four node cluster, quorum is two or three respectively, and losing two nodes will cause a loss of quorum. The loss of quorum results in all services being taken offline.

To resolve the small cluster quorum problem, a quorum disk with a vote count equal to the number of cluster nodes minus one bolsters the quorum enough to enable the cluster to survive with only one remaining node.

The quorum disk daemon (**qdiskd**) runs on each node in the cluster, periodically evaluating its own health and then placing its state information into an assigned portion of the shared disk area. Each **qdiskd** then looks at the state of the other nodes in the cluster as posted in their area of the QDisk partition. When in a healthy state, the quorum of the cluster adds the vote count for each node plus the vote count of the qdisk partition. In the above example, the total vote count is five; one for each node and two for the qdisk partition.

If, on any node, qdiskd is unable to access its shared disk area after several attempts, then **qdiskd** on another node in the cluster attempts to fence the troubled node to return it to an operational state.

## 6.4.2 Additional heuristics

Red Hat added an additional feature to the quorum disk mechanism. Optionally, one or more heuristics can be added to the qdisk configuration. Heuristics are tests performed by the qdisk daemon to verify the health of the node on which it runs. Typical examples are verifications of network connectivity such as the server's ability to ping network routers. Heuristics can also be used to implement network tiebreaker functionality.



## 6.5 Fencing

### 6.5.1 Power Fencing Systems

The power fencing subsystem allows operational cluster nodes to control the power of failed nodes to ensure that they do not access storage in an uncoordinated manner. Most power control systems are network based. They are available from system vendors as add-in cards or integrated into the motherboard. External power fencing devices are also available. These are typically rack or cabinet mounted power switches that can cut the power supply on any given port.

Note that this fencing method requires a working “admin” network connecting to the fence device to successfully trigger the fence action. Fencing devices are recommended to be on the same network that is used for cluster communication.

If the power fencing method uses a remote console (IBM RSA: Remote Supervisor Adapter, Fujitsu iRMC: Integrated Remote Management Controller, HP iLO: integrated lights-out, etc.) extensive testing of the fencing mechanism is recommended. These fencing mechanisms have a short time gap between issuing the “reset” command on the remote console and the actual reset taking place. In a situation when both nodes of a 2-node cluster are trying to fence each other, sometimes the time gap is long enough for both nodes to successfully send the “reset” command before the resets are executed. This results in a power cycle of both nodes.

### 6.5.2 SAN Switch Based Fencing

While it is preferable to employ a power fencing solution for the robustness a system reboot provides, SAN switch fencing is also possible. As with Power Fencing, the need is to protect shared data. SAN switch fencing works by preventing access to storage LUNs on the SAN switch.

### 6.5.3 SCSI Fencing

SCSI-3 persistent reservations can be used for I/O fencing. All nodes in the cluster must register with the SCSI device to be able to access the storage. If a node has to be fenced, the registration is revoked by the other cluster members.

Reference the `fence_scsi(8)` manpage for further details. Please note that the SCSI fencing mechanism requires SCSI-3 write-exclusive, registrants-only persistent reservation as well as support of the preempt-and-abort command on all devices managed or accessed by the cluster. Please contact Red Hat technical support to determine if your software and hardware configuration supports persistent SCSI reservations.

The [How to Control Access to Shared Storage Devices Using SCSI Persistent Reservations with Red Hat Enterprise Linux Clustering and High Availability](#) technical brief discusses this more.



## 6.5.4 Virtual Machine Fencing

VM fencing uses the virtual machine hypervisor to reset or power off a single VM. Please refer to <https://access.redhat.com/kb/docs/DOC-46375> and [SAP note 1552925 - "Linux: High Availability Cluster Solutions"](#) regarding the support status of virtual machine fence devices.

At the time of publication VMware vSphere 4.1+, VMware vCenter 4.1+, VMware ESX 4.1+ and VMware ESXi 4.1+ are supported by RHEL 5.7 and later.

## 6.6 Storage Protection

### 6.6.1 HA-LVM, CLVM

Data consistency must be ensured in all cluster configurations. Logical volume configurations can be protected by the use of HA-LVM or CLVM. CLVM is an extension to standard Logical Volume Management (LVM) that distributes LVM metadata updates to the cluster. The *CLVM DAEMON* (**clvmd**) must be running on all nodes in the cluster and produces an error if any node in the cluster does not have this daemon running. HA LVM imposes the restriction that a logical volume can only be activated exclusively; that is, active on only one machine at a time. Red Hat Enterprise Linux 5.6 introduced the option to use HA-LVM with CLVMD, which implements exclusive activation of logical volumes. Previous releases did implement HA-LVM without CLVMD using LVM tags filtering.

### 6.6.2 GFS

Red Hat's Global File System (GFS) is a POSIX compliant, shared disk cluster file system. GFS lets servers share files with a common file system on a SAN.

With local file system configurations such as ext3, only one server can have access to a disk or logical volume at any given time. In a cluster configuration, this approach has two major drawbacks. First, active/active file system configurations cannot be realized, limiting scale out ability. Second, during a failover operation, a local file system must be unmounted from the server that originally owned the service and must be remounted on the new server.

GFS creates a common file system across multiple SAN disks or volumes and makes this file system available to multiple servers in a cluster. Scale out file system configurations can be easily achieved. During a failover operation, it is not necessary to unmount the GFS file system because data integrity is protected by coordinating access to files so that reads and writes are consistent across servers. Therefore, availability is improved by making the file system accessible to all servers in the cluster. GFS can be used to increase performance, reduce management complexity, and reduce costs with consolidated storage resources. GFS runs on each node in a cluster. As with all file systems, it is basically a kernel module that runs on top of the Virtual File System (VFS) layer of the kernel. It controls how and where the data is stored on a block device or logical volume. By utilizing the Linux distributed lock manager to synchronize changes to the file system, GFS is able to provide a cache-coherent, consistent view of a single file system across multiple hosts.



### 6.6.3 Storage Mirroring

In disaster tolerant configurations, storage mirroring techniques are used to protect data and ensure availability in the event of a storage array loss. Storage mirroring is normally performed in two different ways.

Enterprise storage arrays typically offer a mechanism to mirror all data from one storage array to one or more other arrays. In the case of a disaster, remote data copies can be used. When using array-based-mirroring (also known as SAN-based-mirroring) the cluster nodes need access to all mirrors (usually through multipath configurations). However, only one array is used by the cluster at one time (active array); the other array is used for replication and site failover purposes (passive array). If the active storage array fails, cluster services halt and the cluster must be manually stopped and reconfigured to use passive array.

In Red Hat Enterprise Linux 5.6, Red Hat offers the possibility to create host-based-mirroring configurations with the logical volume manager (LVM). Servers in the cluster are able to assemble independent storage devices (LUNs) on separate storage arrays to a soft-raid logical volume in order to prevent data loss and ensure availability in case of a failure on one of the physical arrays. LVM simultaneously reads and writes to two or more LUNs on separate storage arrays and keeps them synchronized.

## 6.7 OS Dependencies

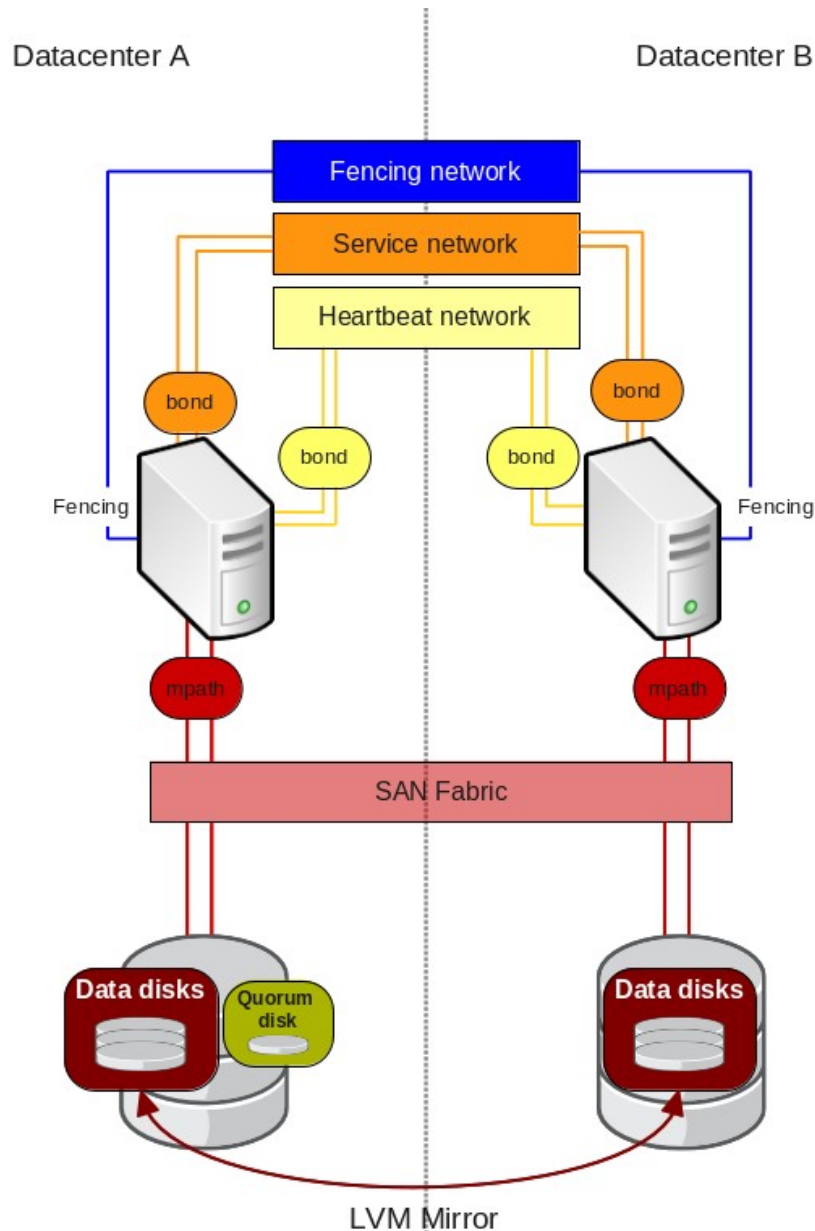
All changes to the operating system environment have to be rolled out to all nodes participating in the cluster. This includes changes to

- configuration files:
  - */etc/lvm/lvm.conf*
  - */etc/cluster/mdadm.conf*
  - */etc/services*
  - */etc/hosts*
  - */etc/multipath.conf*
  - */etc/multipath.binding*
- os packages and patches
- os users and groups
  - home-directories
  - user settings and logon scripts



## 6.8 Stretch Clusters

Red Hat Clustering can be used to provide disaster recovery capabilities in order to minimize service downtime in physical site failure scenarios. Stretch clusters span two sites and have LAN-like latency between sites via site-to-site interlink. Red Hat supports different stretch cluster architectures depending on the the storage infrastructure requirements and data replication technologies as shown in Knowledge Base article “[Support for Red Hat Enterprise Linux Cluster and High Availability Stretch Architectures](#)”. This section will focus on the “Fully Interconnected SAN with LVM Mirroring” use case, described in following diagram:



**Figure 6.8.1: Fully Interconnected SAN with LVM Mirroring**



## 6.8.1 Network infrastructure requirements

In this configuration cluster nodes are distributed between two sites. Both sites must be connected by a network interconnect that provides a LAN-like latency ( $\leq 2\text{ms}$  round trip ping time), and share logical networks. Multicast or broadcast must work between the nodes in the different sites.

## 6.8.2 Storage requirements

This solution requires at least two storage arrays, one at each physical site, with full SAN connectivity between all arrays and all nodes at each site.

## 6.8.3 Quorum in stretch clusters

There are several techniques for quorum management depending on the number and distribution of nodes, these are listed below. A quorum disk used in any of these techniques must be a single disk in the cluster and can not be replicated.

- Two nodes clusters can use two-node mode with fencing delays. A *FENCING LOOP* that causes the nodes to fence each other may occur and must be taken into consideration.
- An iSCSI based quorum disk is recommended for clusters with more than three nodes and can be used in two node clusters.
- For evenly distributed clusters, a tiebreaker node at a third site can be used.
- A Quorum disk or a tiebreaker node can be located in one of the cluster sites. However, manually editing the cluster expected votes is required during failover in site failure cases if tiebreaker node or quorum disk is located in the affected site.

## 6.8.4 Data replication with LVM

LVM mirror is used to synchronously replicate storage between the two arrays. Several points must be considered when configuring LVM mirror in stretch clusters.

- LVM tags based HA-LVM is required in this case to ensure data consistency.
- When creating LVM mirrors, the mirror legs must be created in storage arrays at different physical sites.
- To avoid mirror resynchronization on reboot, disk based mirrorlog must be used. Since RHEL 5.6, mirrorlog can be mirrored itself.





## 6.8.5 Stretch cluster limitations

The use of stretch clusters impose some restrictions when configuring Red Hat Clustering.

- A maximum of two sites are supported. This does not include a third site, which may be used for additional quorum information via a quorum disk.
- CLVMD is not supported in stretch clusters. HA-LVM with tags must be used to ensure data consistency in shared volumes.
- Cluster aware mirroring (cmirror), GFS, and GFS2 are not supported in stretch cluster environments.

## 6.8.6 Site failure management

When a site failure is detected, rgmanager will suspend cluster services until quorum is restored. As seen in section **6.8.3 Quorum in stretch clusters**, there are different techniques that can be applied. So the quorum behavior in a site failure scenario must be carefully analyzed.

- In two node clusters with two-node mode enabled, each node can grant quorum by itself, so the cluster will never lose quorum.
- In the event of a datacenter failure, if the clusters are evenly distributed with the quorum disk at a third site, the nodes at the healthy site will maintain quorum while connectivity to the third site is not lost.
- Site failure scenarios that lead to quorum disk loss (when quorum disk is located in one of the datacenters) setting the expected votes manually will be required to gain quorum.

Once quorum is restored, the administrator must override the node fencing using `fence_ack_manual`. Once manual fencing acknowledgment has been performed, the services running in the failed site will be relocated to the surviving site.

Despite both nodes being quorate, recovery still waits for fencing to complete. This ensures the preservation of data integrity.

## 6.8.7 Site recovery management

When a site failure occurs, LVM mirror replication becomes broken. When a site is recovered, mirror recovery must be done manually. This procedure must be performed while the mirror is not being accessed by a live service. Performing a mirror recovery while the mirror is in use by the cluster or by the cluster services is not supported.





## 6.8.8 Stretch clusters architecture review

Stretch clusters must be carefully designed and implemented to ensure proper behavior in all failure scenarios. Stretch clusters require obtaining a formal review from Red Hat Support as described in knowledge base article “[Architecture Review Process for Red Hat Enterprise Linux High Availability, Clustering, and GFS/GFS2](#)” to ensure that the deployed cluster meets established guidelines.



# 7 Operating System Installation

Reference the *Red Hat Enterprise Linux Installation Guide* for the specific details regarding the acquisition and installation of Red Hat Enterprise Linux. The guide includes information specific to the platform on which the installation takes place (x86, AMD64, Intel® 64 and Itanium), be sure to read the appropriate section for your platform.

Once the platform specific information has been understood and the hardware configuration has been performed to accommodate a cluster, install Red Hat Enterprise Linux 5.6 on the servers using the preferred method.

The installation assistant guides the user through the OS installation. The Red Hat Enterprise Linux Installation Guide provides details regarding each of the screens presented during the installation process.

Refer to the *Installation* section of “Configuring and Managing a Red Hat Cluster” to make sure the required cluster software packages are installed. SAP note 1048303 lists all required software groups and packages that are necessary in order to run SAP software.

## 7.1 OS Installation overview

For a cluster with a local root file system configuration, the following steps must be performed on every cluster node:

1. Install the Red Hat Enterprise Linux 5 operating system
2. Install the required cluster packages
3. Perform the required OS customizations
4. Perform the required network configurations
5. Perform the local storage configuration procedure
6. On one node only, create a cluster configuration file and copy the file to all other cluster members.
7. Start the cluster daemons on all nodes

### 7.1.1 Core Cluster Software Installation

1. Subscribe all cluster nodes to the “RHEL Clustering” channel on RHN or your local RHN Satellite Server
2. Install the required RPMs with the following **yum** command:

```
# yum groupinstall Clustering
```



## 7.2 OS Customizations

### 7.2.1 SAP specific OS customization

Please make sure that all OS customizations listed in [SAP note 1048303 - “Red Hat Enterprise Linux 5.x: Installation and upgrade”](#) have been applied.

### 7.2.2 NTP

The synchronization of system clocks in a cluster becomes infinitely more important when storage is shared among members. System times should be synchronized against a network time server via the Network Time Protocol (NTP) by using the `ntpd` service.

### 7.2.3 ACPI

Please reference the *Configuring ACPI For Use with Integrated Fence Devices* section in “Configuring and Managing a Red Hat Cluster”. As described there, disabling ACPI Soft-Off allows an integrated fence device to shut down a server immediately rather than attempting a clean shutdown.

Soft-Off allows some components to remain powered so the system can be roused from input from the keyboard, clock, modem, LAN, or USB device and subsequently takes longer to fully shutdown. If a cluster member is configured to be fenced by an integrated fence device, disable ACPI Soft-Off for that node. Otherwise, if ACPI Soft-Off is enabled, an integrated fence device can take several seconds or more to turn off a node since the operating system attempts a clean shutdown. In addition, if ACPI Soft-Off is enabled and a node panics or freezes during shutdown, an integrated fence device may not be able to power off the node. Under those circumstances, fencing is delayed or unsuccessful.

Use the following commands to switch off ACPI Soft-Off:

```
# service acpid stop
# chkconfig acpid off
```



## 7.2.4 Firewall

If use of a local firewall on the cluster nodes is intended, the specific IP ports for the following services must be enabled in order to accommodate RHEL HA Add-On communication requirements. The services and ports are listed in the following table.

Service	IP Ports
openais	5404, 5405
ricci	11111
dlm	21064
ccsd	50006, 50007, 50008, 50009

**Table 7.2.4.1: Service Ports**

## 7.3 Network Configuration

In a cluster configuration, the configuration of the cluster interconnect is extremely important. The interconnect is responsible for all internal cluster communication. With a clustered file system, all distributed locking messages are routed through the cluster interconnect. As such, it is highly recommended that the network be reliable and high speed.

### 7.3.1 Public/Private Networks

At least two network interfaces are recommended for clustering. The reason for this is to separate cluster traffic from all other network traffic. Availability and cluster file system performance is dependent on the reliability and performance of the cluster communication network (private network).

Therefore, all public network load must be routed through a different network (public network).

### 7.3.2 Bonding

In high availability configurations, at least the private or cluster interconnect network setup, preferably both, must be fully redundant. Network Interface Card (NIC) bonding is the only method to provide NIC failover for the cluster communication network.



### 7.3.3 Hosts file

The `/etc/hosts` file for each cluster member should contain an entry defining `localhost`. If the external host name of the system is defined on the same line, the host name reference should be removed. For example, if `/etc/hosts` contains a line like this

```
127.0.0.1 localhost.localdomain localhost foo.bar.com foo
```

please change it to this

```
127.0.0.1 localhost.localdomain localhost
```

Additionally, each `/etc/hosts` file should define the local interconnect of each cluster member.

## 7.4 Storage Configuration

### 7.4.1 Multipathing

Storage hardware vendors offer different solutions for implementing multipath failover capability. This document focuses on the generic multipath device mapper approach.

### 7.4.2 Device Mapper Multipath

The device mapper multipath plugin (DM multipath) provides greater reliability and performance by using path failover and load balancing. In HA scenarios, cluster servers can use multiple paths to the shared storage devices. Normally these devices are presented as multiple device files (`/dev/sdXX`)

DM-Multipath creates a single device that routes I/O to the underlying devices according to the multipath configuration. It creates kernel block devices (`/dev/dm-*`) and corresponding block devices (with persistent names) in the `/dev/mapper` directory.

The multipath configuration file `/etc/multipath.conf` can also be used to set storage specific attributes. These multipath specific settings are usually obtained from the storage vendor and typically supersede the default settings. Please consult your storage hardware vendor for the correct and supported multipath configuration.



The mapping of LUN wwid's to user friendly device names should be enabled in */etc/multipath.conf* by setting these two variables within the defaults section:

```
defaults {  
    user_friendly_names yes  
    bindings_file /etc/multipath.bindings  
}
```

The bindings file only contains the alias name and the LUN wwid in the format:

```
<alias1> <wwid1>  
<alias2> <wwid2>
```

[ ... output truncated ... ]

If binding is configured but the bindings file doesn't exist, it is created during next system boot.

The bindings have to be consistent across all cluster nodes. Distribute *multipath.conf* and *multipath.bindings* to all nodes participating in the cluster.

Make sure to use the correct driver, path priorities and load balancing algorithm for your SAN storage. Some storage arrays use preferred paths. Even though the underlying storage LUN can be accessed through all available FC paths only certain paths are optimized for maximum performance. Red Hat Linux 5.6 supports the ALUA protocol to automatically detect and use performance optimized paths.

The command **multipath -ll** lists all multipath devices, storage LUNs, FC paths and path status. “Enabled” means that the path is available but currently not used for I/O (backup / failover path). “Active” means that this path is in use and receiving I/O requests.

### 7.4.3 LVM

In RHEL HA Add-On, LVM managed shared storage can be controlled by High Availability resource manager agents for LVM (HA-LVM) or the clustered logical volume manager daemon (clvmd/CLVM).

The example configuration in this paper uses HA-LVM because the services managed by the cluster software (SAP DB, SAP instances) have dedicated volume groups with dedicated physical volumes. This avoids concurrent access to logical volumes originating from multiple nodes.



### 7.4.3.1 LVM Configuration

The LVM configuration file `/etc/lvm/lvm.conf` must be modified to enable the use of HA-LVM.

1. By default, the LVM commands scan all devices found directly in the `/dev` path. This is insufficient in dm-multipath configurations. There are two ways to enable multipath devices for LVM. The easiest is to modify the scan array in the configuration file as follows:

```
scan = [ "/dev/mapper" ]
```

2. In order to allow cluster members to activate logical volumes through the HA-LVM resource agent it is necessary to maintain a `volume_list` in `lvm.conf`

```
# If volume_list is defined, each LV is only activated if there is a
# match against the list.
# "vgname" and "vgname/lvname" are matched exactly.
# "@tag" matches any tag set in the LV or VG.
# "@*" matches if any tag defined on the host is also set in the LV or
VG

volume_list = [ "vg0", "@ls3121", "vg_cor_scs/lv_cor_scs",
"vg_cor_java/lv_cor_java", "vg_cor_db/lv_cor_db",
"vg_cor_sap/lv_cor_sap", "vg_cor_ers/lv_cor_ers",
"vg_cor_scs/lv_cor_scs@ls3110hb",
"vg_cor_java/lv_cor_java@ls3110hb", "vg_cor_db/lv_cor_db@ls3110hb",
"vg_cor_sap/lv_cor_sap@ls3110hb", "vg_cor_ers/lv_cor_ers@ls3110hb",
"vg_cor_scs/lv_cor_scs@ls3121hb",
"vg_cor_java/lv_cor_java@ls3121hb", "vg_cor_db/lv_cor_db@ls3121hb",
"vg_cor_sap/lv_cor_sap@ls3121hb", "vg_cor_ers/lv_cor_ers@ls3121hb" ]
```

Modifications to `lvm.conf` have to be compiled into the kernel boot package:

```
# new-kernel-pkg --mkinitrd --initrdfile=/boot/initrd-halvm-`uname -r`.img
--install `uname -r`
```

Please reference the [LVM Administrators Guide](#) for more detailed information on using LVM.

### 7.4.3.2 Volume Configuration

In this configuration, several storage LUNs were defined to store application data. Each LUN is used for one logical volume group and one logical volume. The following steps were performed to create the logical volume `/dev/mapper/vg_cor_db-lv_cor_db`

```
# pvcreate /dev/mapper/mpatha
# vgcreate vg_cor_db /dev/mapper/mpatha
# lvcreate -n lv_cor_db -L 36G vg_cor_db
```



## 7.5 Cluster Core Configuration

The cluster configuration file, `/etc/cluster/cluster.conf` (in XML format), for this cluster contains the following outline:

```
<?xml version="1.0"?>
<cluster>
  <cman/>
  <totem/>
  <quorumd/>
  <fence_daemon/>
  <clusternodes>
    <clusternode/>

    [ ... output abbreviated ... ]

    <clusternode/>
  </clusternodes>
  <fencedevices>
    <fencedevice/>

    [ ... output abbreviated ... ]

    <fencedevice/>
  </fencedevices>
  <rm>
    <!-- Configuration of the resource group manager -->
  </rm>
</cluster>
```

The cluster configuration can be created in three different ways:

1. **Conga** - A web based configuration interface
2. **system-config-cluster** - The local cluster configuration GUI
3. **vi** - File editor

Unfortunately some configuration details can only be defined using a common file editor such as **vi**.





It is recommended to create the initial cluster configuration file with the help of a GUI based tool (**system-config-cluster**) and later perform all necessary modifications by hand. To manually alter the configuration within a running cluster, the following steps must be performed:

1. Increment the **config\_version** attribute within the **<cluster>** tag:

```
<cluster config_version="X"/>
```

2. Update the ccs cluster configuration:

```
# ccs_tool update /etc/cluster/cluster.conf
```

This distributes the new configuration to all members of the cluster.

The **<cluster>** tag should define the following attributes:

Attribute	Description
config_version	Version number of the configuration
Name	The name of the cluster

**Table 7.5.1: Cluster Tag Attributes**

## 7.5.1 CMAN / OpenAIS

The OpenAIS daemon, **aisexec**, is started and configured by CMAN. Typically, all work is performed within the cman init script. The following command can be used to start the **aisexec** daemon:

```
# service cman start
```

The CMAN/OpenAIS portion within the cluster configuration file is defined within the **<cman>** tag. The following attributes should be taken into consideration:

Attribute	Description
expected_votes	Number of votes used to calculate the quorum
two_node	Special configuration option for 2-node clusters which ignores quorum requirements

**Table 7.5.1.1: CMAN Tag Attributes**



## 7.5.2 Qdisk

If the use of a quorum device is intended, the following steps must be performed:

1. Format a shared disk partition as quorum disk:  

```
# mkqdisk -c <device> -l <label>
```
2. Add a **<quorumd>** configuration tag to the cluster configuration file.
3. Optionally define helpful heuristics for qdiskd verification purposes

The **<quorumd>** tag should define the following attributes:

Attribute	Description
interval	The frequency of read/write cycles, in seconds.
tko	The number of cycles a node must miss in order to be declared dead.
votes	The number of votes the quorum daemon advertises to CMAN when it has a high enough score.
log_level	Controls the verbosity of the quorum daemon in the system logs. 0 = emergencies; 7 = debug.
log_facility	Controls the syslog facility used by the quorum daemon when logging. For a complete list of available facilities, see <code>syslog.conf(5)</code> . The default value for this is "daemon"
min_score	Absolute minimum score to consider one's self "alive". If omitted, or set to 0, the default function " $\text{floor}((n+1)/2)$ " is used, where $n$ is the total of all of defined heuristic scores. This must never exceed the sum of the heuristic scores, or else the quorum disk never becomes available.
device	The device the quorum daemon uses. This device must be the same on all nodes. label Overrides the device field if present. If specified, the quorum daemon reads <code>/proc/partitions</code> and search for qdisk signatures on every block device found, comparing the label against the specified label. This is useful in configurations where the block device name differs on a per-node basis.
master_wins	Controls the voting behavior of qdiskd. If set to a value of 1, then only the qdiskd master provides votes to CMAN. This ensures that the qdiskd master automatically wins in a fence race. See the <code>qdisk(5)</code> man page for more information about this attribute.

**Table 7.5.2.1: Quorumd Tag Attributes**



In the case of a split brain situation, heuristics can be used to identify the cluster partition that is best to survive. Heuristics can be defined by adding a **<heuristic>** tag within the **<quorumd>** tag.

The **<heuristic>** tag should define the following attributes:

Attribute	Description
program	The program used to determine if this heuristic is alive. This can be anything executable by <b>/bin/sh -c</b> . A return value of zero indicates success; anything else indicates failure.
score	The weight of this heuristic. Be careful when determining scores for heuristics. The default score for each heuristic is 1.
interval	The frequency (in seconds) at which we poll the heuristic. The default interval for every heuristic is 2 seconds.
tko	The number of heuristic test failures before a node is considered DOWN and its score is removed. The default tko for each heuristic is 1, which can be inadequate for actions such as 'ping'.

**Table 7.5.2.2: Heuristic Tag Attributes**

For more detailed information, refer to the **qdisk** man page. If device mapper multipath is used together with **qdiskd**, the values for **tko** and **interval** must be carefully considered. In the example case of a path failover, all storage I/O is queued by the device mapper module. The **qdisk** timeout must be adapted to the possible device mapper's queuing time. A detailed analysis on Quorum Disk configuration can be found in the technical brief "[How to Optimally Configure a Quorum Disk in Red Hat Enterprise Linux Clustering and High-Availability Environments](#)".

## 7.5.3 Fencing

The fencing configuration consists of two parts. The first is the configuration of the fencing daemon (**fenced**) itself. The second is the configuration of the fencing agents that the daemon uses to fence each cluster node.

The fencing daemon is configured by adding the **<fence\_daemon>** tag within the cluster configuration file. The following attributes should be considered:

Attribute	Description
post_join_delay	Post-join delay is the number of seconds the daemon waits before fencing any victims after a node joins the domain.
post_fail_delay	Post-fail delay is the number of seconds the daemon waits before fencing any victims after a domain member fails.

**Table 7.5.3.1: Fence\_daemon Tag Attributes**



The fencing agents used for each cluster node must be configured within the `<fencedevices>` tag. For each fencing device, a `<fencedevice>` tag within the `<fencedevices>` tag must be defined. The `<fencedevice>` tag should at minimum define the agent and name attributes.

The fencing system supports multiple layers of fence devices making it is possible to configure more than one way to fence a cluster node. A fencing attempt succeeded if all fence actions (fence devices) that belong to the same fence-level (method) were successfully executed.

For further information about the different configuration options of all fencing agents, reference the man pages of the desired fencing agent.

## 7.5.4 Cluster Nodes

The configuration of the cluster nodes is controlled by `<clusternode>` tags within an encapsulating `<clusternodes>` tag. The basic cluster node configuration should contain at least the following attributes:

Attribute	Description
name	The name of the host
nodeid	The id of the cluster node
votes	The number of quorum votes for this node

**Table 7.5.4.1: Clusternodes Tag Attributes**

Within the `<clusternode>` tag, the methods used to fence the node must be defined. All fencing mechanism are encapsulated within the `<fence>` tag. Each fencing mechanism is defined by the `<method>` tag.

Please refer to the man pages of fence as well as the man pages for the chosen fencing mechanisms for further details.

8. On every cluster node, start the cman init script:

```
# service cman start
```

9. To verify the changes have been propagated, the version number and cluster status can be viewed on any node at any time using `cman_tool`.

```
# cman_tool status
```

10. The state of all cluster nodes can be viewed with the following command:

```
# cman_tool nodes
```



# 8 SAP Installation

## 8.1 SAP Architecture

Following the established SAP documentation is highly recommended:

- SAP Installation Guide  
<http://service.sap.com/instguides>
- SAP Technical Infrastructure Guide  
<https://www.sdn.sap.com/irj/sdn/ha>

## 8.2 Virtual IP Addresses

SAP NetWeaver is typically installed via the graphical installation tool **sapinst**. Before beginning the installation, determine which IP addresses and host names are preferred for use during the SAP installation. First, each node requires a static IP address and an associated host name. This address is also referred to as the physical IP address. Second, each database and SAP instance requires a virtual IP address / host name. The virtual addresses must not be configured at the operating system level because they are under the control of the Clustering. Those addresses are referred to as the virtual IP addresses. The virtual IP address and virtual hostname guarantee that a database or SAP instance is always accessible under the same name no matter which physical cluster node currently hosts the service.

Local dialog instances, which are not part of the cluster, use a virtual host name as an alias to the physical host name so those SAP instances are not failed over by RHEL HA Add-On. The enqueue replication instances do not need IP addresses because no connections are established with them. The virtual host name is used to start the instances manually via the **sapstart** command and to distinguish their profile names from physical host names.

Edit the `/etc/hosts` file on all nodes and add the virtual host names and their associated IP addresses or add them to your DNS server. Additionally, add any other cluster relevant host name and address (e.g., the physical host names or addresses of the nodes) to `/etc/hosts` so the DNS server is no longer a possible single point of failure.

## 8.3 File Systems

The file systems for our scenario must be prepared before installing SAP NetWeaver. File systems are set up locally, on shared storage with local file system type (e.g., Ext3), and on a highly available NFS server.



### 8.3.1 Local File Systems

Directories such as `/usr/sap`, `/sapmnt`, and `/oracle` can be created locally on each node. The linking directory `/usr/sap/<SID>/SYS` can also reside locally because it contains only links to `/sapmnt/<SID>`. After initial installation of SAP copy the `/usr/sap/<SID>/SYS` directory to all cluster nodes. The directory `/usr/sap/tmp` should also be locally on every cluster member.

Specific directories for SAP agents such as `/usr/sap/ccms`, `/usr/sap/<SID>/ccms` or `/usr/sap/SMD` must be configured according to your SAP landscape.

### 8.3.2 Shared Storage File Systems

The instance directories `/usr/sap/<SID>/<InstanceNo>` must be set up on shared storage, so that these directories are able to perform a switchover triggered by the cluster software. The database directory `/oracle/<SID>` and its sub-directories containing the DBMS executables, datafiles, logfiles and logarchives must also reside on shared storage while `/oracle/client` should be local to each cluster node.

Follow the database file system configuration recommendations from the SAP installation guide. It is recommended to have physically different mount points for the program files and for *origlog*, *mirrlog*, log archives and each sapdata.

NOTE: The configuration process gets more complex when multiple database instances of the same type run within the cluster. The program files must be accessible for every instance. The mounts from shared storage must be added to the cluster configuration as file system resources to the failover service.

### 8.3.3 NFS Mounted File Systems

The `/sapmnt/<SID>` file system should reside on a high available NFS server or NFS exporting storage array to be available for additional application server outside the cluster. The transport directory `/usr/sap/trans` should also be exported via NFS according to the SAP landscape.

It is possible but not recommended or supported to run an NFS server within the cluster. This leads to a re-mount scenario in which the cluster node exporting the NFS filesystems re-mounts the same NFS exports. In low memory situations, the NFS server and client can negatively impact system stability.

## 8.4 Before Starting the SAP Installation

Before installing SAP NetWeaver, mount all the necessary filesystems (either through the cluster or manually). Be conscious of the overmount effect by mounting the hierarchically highest directories first.

Enable the virtual IP address (either through the cluster or manually). This is necessary because SAPinst starts the newly created instance during its post-processing.



## 8.5 Installation with sapinst

When starting the SAP installation tool (**sapinst**), specify the virtual host name.

```
sapinst SAPINST_USE_HOSTNAME=<virtual hostname>
```

For each SAP and database instance choose the installation option "High-Availability System" as described in the SAP installation guide.

## 8.6 Installation Post-Processing

### 8.6.1 Users, Groups and Home Directories

Create users and groups on the second node as they were created by the SAP installation on the first node. Use the same user and group IDs.

Depending on the Installation Master CD that was used for the SAP installation, the login profiles for the SAP administrator user (<sid>adm) and the database administrator user could differ. In older and non-HA installations, the user login profiles look similar to `.sapenv_hostname.csh`.

Using the host name in the user login profiles is a problem in an HA environment. By default, the profiles `.login`, `.profile` and `.cshrc` search for two types of user login profiles: first for the one including the local host name (e.g., `.dbenv_hostname.csh`) and then for a version without the host name included. The latest versions of the InstMaster CDs install both versions of the user login profiles. This could lead to some confusion for the administrator with regard to which version is used in which case. The removal of all user login profiles (that include a host name in the file name) is recommended. Do this for both the SAP administrator, <sid>adm, as well as the database administrator user.

### 8.6.2 Synchronizing Files and Directories

Copy `/etc/services` or its values that were adjusted by **sapinst** (see SAP related entries at the end of the file) to all other nodes.

Copy the files `/etc/oratab` and `/etc/orainst.loc` to the other nodes.

Copy the directory `/oracle/client/` to all cluster nodes. Remember that in case of a client-patch all nodes have to be updated.

The most important SAP profile parameter for a clustered SAP system is **SAPLOCALHOST**. After the installation with **sapinst**, ensure that all SAP instance profiles contain this parameter. The value of the parameter must be the virtual host name specified during the installation.

As a general recommendation, the SAP parameter **es/implementation** should be set to "std" in the SAP `DEFAULT.PFL` file. See SAP Note 941735. The SAPInstance resource agent cannot use the `AUTOMATIC_RECOVERY` function for systems that have this parameter set to "map".

In the START profiles, the parameter **SAPSYSTEM** must be set (default since 7.00).



### 8.6.3 SAP Release-specific Post-processing

For improved SAP hardware key determination in high-availability scenarios it might be necessary to install several SAP license keys based on the hardware keys of each cluster node. Please see SAP Note 1178686 for more information.

For SAP kernel release 6.40, follow the instructions of SAP Note 877795.

For SAP kernel release 6.40, update the SAP kernel to at least patch level 208.

When using a SAP kernel 6.40, please read and implement the actions from the section "Manual post-processing" from SAP Note 995116.

### 8.6.4 Before Starting the Cluster

In some cases sapinst doesn't start the freshly installed instance and leaves an empty work directory (`/usr/sap/<SID>/<Instance><Number>/work`) which results in a monitoring error of the SAPInstance resource agent.

In that case the instance must be started manually in order for the correct entries to be written to the work directory. After a manual shutdown of the instances, the cluster agent can be used. Remember that the virtual IP addresses for the SAP instances you wish to start must be active. They can be started manually (e.g., with the Linux command `ip`) and then stopped again after shutting down the SAP instances.

Shutdown the SAP instance and integrate it into the cluster by creating a SAPInstance resource (see chapter 7.4.4).

## 8.7 Enqueue Replication Server

Follow the instructions of the official SAP Library to setup an enqueue replication server:

[http://help.sap.com/saphelp\\_nw73/helpdata/en/47/e023f3bf423c83e10000000a42189c/frame\\_set.htm](http://help.sap.com/saphelp_nw73/helpdata/en/47/e023f3bf423c83e10000000a42189c/frame_set.htm)





# 9 Cluster Configuration

## 9.1 Cluster Resources

There are many types of configurable cluster resources. Reference the Adding a Cluster Service to the Cluster section of Configuring and Managing a Red Hat Cluster for more information.

The following resource types are defined to provide the high availability functionality for SAP

IP	manages virtual IP addresses
LVM	manages LVM activations
FS	mounts file-systems
SAPDatabase	starts / stops / monitors the Database
SAPInstance	starts / stops / monitors a SAP instance (ABAP or java)



## 9.2 Basic rgmanager Configuration

The resource group manager is configured within the cluster configuration file `/etc/cluster/cluster.conf`. The configuration is encapsulated within the `<rm>` tag. The resource manager configuration has the following basic layout:

```
<rm>
  <failoverdomains>
    <failoverdomain/>

[ ... output abbreviated ... ]

    <failoverdomain/>
  </failoverdomains>
  <resources>
    <resource/>

[ ... output abbreviated ... ]

    <resource/>
  </resources>
  <service>

[ ... output abbreviated ... ]

    </service>
    <service>

[ ... output abbreviated ... ]

    </service>
    <events>
      <event/>

[ ... output abbreviated ... ]

    </events>
</rm>
```



The following `<rm>` attributes can be defined:

Attribute	Description
log_level	The log level is number from 0..7, where 7 is 'debug' and 0 is 'emergency only'. The default value is 4.
log_facility	Log facility name, such as daemon, local4, or similar. The default value is daemon.
central_processing	The central_processing option is used to activate the event mechanism. Central_processing is needed to enable the hard and soft service dependencies and the follow_service dependency.

**Table 9.2.1: Rm Tag Attributes**

## 9.3 Failover Domains

The cluster can be divided into logical subsets of cluster nodes. A failover domain is a subset of members to which a service can be bound.

Failover domains are configured in the cluster configuration file. The following example outlines the basic configuration schema:

```
<rm>
  <failoverdomains>
    <failoverdomain name="ls3110" ordered="1" restricted="1">
      <failoverdomainnode name="ls3110hb" priority="1"/>
      <failoverdomainnode name="ls3121hb" priority="2"/>
    </failoverdomain>
    <failoverdomain name="ls3121" ordered="1" restricted="1">
      <failoverdomainnode name="ls3121hb" priority="1"/>
      <failoverdomainnode name="ls3110hb" priority="2"/>
    </failoverdomain>
  </failoverdomains>
</rm>
```

The example above creates two failover domains. Services running on failover domain `ls3110` prefer to run on node `ls3110hb` while services running on failover domain `ls3121` prefer to run on node `ls3121hb`. Each cluster node can be part of multiple failover domains.



The following configuration attributes can be set to define the failover rules:

Attribute	Description
restricted	Services bound to the domain can only run on cluster members which are also members of the failover domain. If no members of the failover domain are available, the service is placed in the stopped state.
unrestricted	Services bound to this domain can run on all cluster members, but run on a member of the domain whenever one is available. This means that if a service is running outside of the domain and a member of the domain comes online, the service migrates to that member.
ordered	The order specified in the configuration dictates the order of preference of members within the domain. The highest-ranking member of the domain runs the service whenever it is online. This means that if member A has a higher-rank than member B, the service migrates to A if it was running on B if A transitions from offline to online.
unordered	Members of the domain have no order of preference; any member may run the service. Services always migrate to members of their failover domain whenever possible, however, in an unordered domain.
nofailback	Enabling this option for an ordered failover domain prevents automated fail-back after a more-preferred node rejoins the cluster.

**Table 9.3.1: Failover Attributes**

See the [Configuring a Failover Domain](#) section of the [Cluster Administration Guide](#) for more information.



## 9.4 Cluster Resources and Services

There are many types of cluster resources that can be configured. Resources are bundled together to highly available services while a service consists of one or more cluster resources. The database service for example consists of these resources:

- virtual IP address (IP resource)
- volume groups (LVM resource)
- filesystems for DB executables, datafiles, logs, etc. (FS resource)
- database application start/stop/monitor (SAPDatabase resource)

Resources can be assigned to any cluster service (resource groups). Once associated with a cluster service, it can be relocated by the cluster transition engine if it deems it necessary, or manually through a GUI interface, a web interface (conga) or via the command line. If any cluster member running a service becomes unable to do so (e.g., due to hardware or software failure, network/connectivity loss, etc.), the service with all its resources are automatically migrated to an eligible member (according to failover domain rules).

Reference the Adding a Cluster Service to the Cluster section of “Configuring and Managing a Red Hat Cluster” for more information.

Highly available cluster services are configured within the <service> tag. Consider defining the following attributes:

Attribute	Description		
name	The name of the service or resource group		
domain	The failover domain associated with this service		
autostart	If set to yes, the service automatically starts after the cluster forms a quorum. If set to no, this resource group starts in the 'disabled' state after the cluster forms a quorum. Default is 1.		
exclusive	If set, this resource group only relocates to nodes which run no other resource groups		
recovery	This currently has three possible options:		
		restart	tries to restart failed parts of this resource group locally before attempting to relocate (default)
		relocate	does not bother trying to restart the service locally
		disable	disables the resource group if any component fails
	If a resource has a valid "recover" operation and can be recovered without a restart, it is recovered.		

**Table 9.4.1: Service Tag Attributes**



## 9.4.1 SAP Resources

The following resource types are defined to provide the high availability functionality for SAP.

### 9.4.1.1 IP

The ip resource defines an ipv4 or ipv6 network address. The following attributes can be defined:

Attribute	Description
address	IPv4 or IPv6 address to use as a virtual IP resource
monitor_link	Enabling this causes the status verification to fail if the link on the NIC to which this IP address is bound is not present.

**Table 9.4.1.1: Ip Resource Attributes**

### 9.4.1.2 LVM

The lvm resource controls the availability of a logical volume. The configurable resource attributes are:

Attribute	Description
name	A symbolic name for the file system resource - only as reference within the cluster configuration (used when assigning to a service)
vg_name	The name of the volume group
lv_name	The name of the logical volume

**Table 9.4.1.2: LVM Resource Attributes**



### 9.4.1.3 FS

The fs resource defines a standard local file system mount; i.e., a non clustered or otherwise shared file system. The following attributes can be defined:

Attribute	Description
name	A symbolic name for the file system resource - only as reference within the cluster configuration (used when assigning to a service)
mountpoint	Path within file system hierarchy at which to mount this file system
device	Block device, file system label, or UUID of file system
fstype	File system type. If not specified, mount(8) attempts to determine the file system type.
force_unmount	If set, the cluster kills all processes using this file system when the resource group is stopped. Otherwise, the unmount fails, and the resource group is restarted. It is necessary to set this to „1“ for all SAP instance directories and can be set for all SAP and DB filesystems. DEFAULT: 0
options	Provides a list of mount options. If none are specified, the NFS file system is mounted -o sync.
self_fence	If set and unmounting the file system fails, the node immediately reboots. Generally, this is used in conjunction with force_unmount support, but it is not required.
force_fsck	If set, the file system is verified. This option is ignored for non-journalled file systems such as ext2.

**Table 9.4.1.3: Fs Resource Attributes**

### 9.4.1.4 SAPInstance

Within SAP instances there can be several services. Typically, services are defined in the START profile of the related instance (Note: with SAP Release 7.10, the START profile content was moved to the instance profile). Not all of those processes are worth monitoring by the cluster. For instance, failover of the SAP instance would not be preferred if the central syslog collector daemon failed.

SAP processes monitored by the SAPInstance resource agent are:

- disp+work
- msg\_server
- enserv
- enrepserver
- jcontrol
- jstart



A SAP instance without any of these processes can not be managed with the SAPInstance resource agent.

For example: An environment with a standalone gateway instance or a standalone web dispatcher instance which fails to work with this resource agent. The next version of the agent can have a parameter that could be used to select which services should be monitored. However, this does not mean that a SAP web dispatcher cannot be included in another SAP instance that uses one of the monitored services (e.g., a SCS instance running a msg\_server and a enservr). In this case, the web dispatcher is started and stopped (together with the other services) by the cluster. The web dispatcher is then not monitored, meaning a hung or dead sapwebdisp process does not cause a failover of the entire SAP instance. However, that may be the desired behavior.

All operations of the SAPInstance resource agent are performed by using the SAP startup framework called SAP Management Console, or sapstartsrv, that was introduced with SAP kernel release 6.40. Reference additional information regarding the SAP Management Console in SAP Note 1014480.

Using this framework defines a clear interface for the cluster heartbeat and how it views the SAP system. The monitoring options for the SAP system are far superior than other methods such as monitoring processes with the ps command for or pinging the application.

**sapstartsrv** uses SOAP messages to request the status of running SAP processes. As such, it can request status directly from the process itself, independent from other issues that can exist at the time.

**sapstartsrv** has four status states:

- GREEN = everything is fine
- YELLOW = something is wrong, but the service is still working
- RED = the service does not work
- GRAY = the service has not been started

The SAPInstance resource agent interprets GREEN and YELLOW as acceptable, meaning minor problems are not reported to the cluster. This prevents the cluster from performing an unwanted failover. The statuses RED and GRAY are reported as NOT\_RUNNING to the cluster. Depending on the status the cluster expects from the resource, it performs a restart, a failover, or does nothing.

To debug problems with the SAPInstance resource agent try starting, stopping and monitoring the SAP instance in the same manner as the resource agent would do it. As <sid>adm execute:

```
$ sapcontrol -nr <instance number> -function Start
```

to start the SAP instance

```
$ sapcontrol -nr <instance number> -function Stop
```

to stop the SAP instance

```
$ sapcontrol -nr <instance number> -function GetProcessList
```

to return the current status of the SAP instance





If **sapcontrol** returns a network error the **sapstartsrv** service is not running. The resource agent always tries to restart **sapstartsrv** in case the startup framework is not running.

The SAPInstance resource can be configured with these attributes:

Attribute	Description
InstanceName	<p>The full qualified SAP instance name in the format: &lt;SID&gt;_&lt;INSTANCE&gt;_&lt;VHOST&gt; e.g. COR_DVEBMGS02_corpas</p> <p>This is typically the name of the SAP instance profile</p>
DIR_EXECUTABLE	<p>The full path to sapstartsrv and sapcontrol. Specify this parameter if the SAP kernel directory location has been changed after the default SAP installation. DEFAULT: <i>/usr/sap/&lt;SID&gt;/&lt;INSTANCE&gt;/exe</i> or <i>/usr/sap/&lt;SID&gt;/SYS/exe/run</i></p>
DIR_PROFILE	<p>The full path to the SAP profile directory. Specify this parameter, if you have changed the SAP profile directory location after the default SAP installation. DEFAULT: <i>/usr/sap/&lt;SID&gt;/SYS/profile</i></p>
START_PROFILE	<p>The full path and name of the SAP START profile. Specify this parameter if the name of the SAP START profile was changed after the SAP installation. In SAP release 7.1 the Instance Profile has to be specified because this version of SAP doesn't use a START profile. DEFAULT: <i>\$DIR_PROFILE/START_&lt;INSTANCE&gt;_&lt;VHOST&gt;</i></p>
START_WAITTIME	<p>The time in seconds before a monitor operation is executed by the resource agent. If the monitor returns SUCCESS, the start is handled as SUCCESS. This is useful for resolving timing issues with the J2EE-AddIn instance. Typically, the resource agent waits until all services are started and the SAP Management Console reports a GREEN status. A double stack installation (ABAPJava AddIn) consists of an ABAP dispatcher and a JAVA instance. Normally, the start of the JAVA instance takes longer than the start of the ABAP instance. For a JAVA Instance, one may need to configure a much higher timeout for the start operation of the resource. The disadvantage would be that the discovery of a failed start</p>



Attribute	Description
	<p>by the cluster takes longer.</p> <p>In some cases, it may be more important that the ABAP part of the instance is up and running. A failure of the JAVA add-in does not cause a failover of the SAP instance. Actually, the SAP MC reports a YELLOW status if the JAVA instance of a double stack system fails. From the perspective of the resource agent, a YELLOW status doesn't trigger a cluster reaction Setting <code>START_WAITTIME</code> to a lower value causes the resource agent to verify the status of the instance during a start operation after that time. When it would normally wait for a GREEN status, it now reports SUCCESS in the case of a YELLOW status after the specified time. This is only useful for double stack systems.</p> <p>DEFAULT: 3600</p>
AUTOMATIC_RECOVER	<p>The SAPInstance resource agent attempts to recover a failed start attempt automatically one time. This is accomplished by killing any running instance processes and executing <code>cleanipc</code>.</p> <p>Sometimes a crashed SAP instance leaves some processes and/or shared memory segments behind. Setting this option to true attempts to remove the processes and shared memory segments during a start operation, reducing administrator labor.</p> <p>DEFAULT: false</p>
PRE_START_USEREXIT, POST_START_USEREXIT, PRE_STOP_USEREXIT, POST_STOP_USEREXIT	<p>The fully qualified path to a script or program which should be executed before/after a resource is started/stopped. SAP systems often required additional software run on the same server. That can be monitoring software or software for some interfaces the SAP system uses. Those programs can include by writing a new OCF resource agent to be integrated into the cluster. However, sometimes writing a resource agent is too much effort. With the provided userexits, custom scripts can be easily included into the cluster that do not follow the OCF standard. The returncode of the scripts are not used by the SAPInstance resource agent. The call of userexit is synchron, meaning the time the script requires is going into the timeout of the start/stop operation defined for the SAPInstance resource. If the userexit-script hangs, SAP may not be started.</p> <p>DEFAULT: empty</p>

**Table 9.4.1.4: SAP Instance Resource Attributes**

See <http://linux-ha.org/doc/man-pages/re-ra-SAPInstance.html> for the full list of attributes.



### 9.4.1.5 SAPDatabase

The purpose of the resource agent is to start, stop, and monitor the database instance of an SAP system. Together with the RDBMS system, it also controls the related network service for the database (such as the Oracle Listener or the MaxDB xserver). The resource agent expects a standard SAP installation and therefore requires less parameters.

The monitor operation of the resource agent can test the availability of the database by using SAP tools (**R3trans** or **jdbconnect**). With that, it ensures that the database is truly accessible by the SAP system.

After an unclean exit or crash of a database, require a recover procedure to restart can be required. The resource agent has a procedure implemented for each database type. If preferred, the attribute `AUTOMATIC_RECOVER` provides this functionality.

Attribute	Description
SID	The unique SAP system identifier (e.g. COR)
DBTYPE	The RDBMS System (either: ORA, DB6 or ADA)
DIR_EXECUTABLE	The full qualified path to the SAP kernel. The resource agent requires the <b>startdb</b> and the <b>R3trans</b> executables. For that reason, the directory with the SAP kernel must be accessible to the database server at any given time. Specify this parameter if the SAP kernel directory location was changed after the default SAP installation.
NETSERVICENAME	The Oracle TNS listener name (DEFAULT: LISTENER)
DBJ2EE_ONLY	If no ABAP stack is installed in the SAP database, set this to true. Non ABAP systems cannot be monitored using R3trans. That parameter shifts the monitoring method to jdbconnect. DEFAULT: false Note that some files of the SAP j2ee instance have to be mounted / accessible for this operation to work! This may lead to problems as the SAP instance usually isn't mounted yet during DB startup!
JAVA_HOME	This is required only if the DBJ2EE_ONLY parameter is set to true. Enter the path to the Java SDK used by the SAP WebAS Java. Set this parameter if the environment variable JAVA_HOME is not set for the root user, or points to another directory than that of the JAVA_HOME for the <sid>adm user. DEFAULT: \$JAVA_HOME
STRICT_MONITORING	Controls how the resource agent monitors the database. If true, it uses SAP tools to test the connection to the database. Not for use with Oracle as it results in unwanted failovers in the case of a stuck archiver. DEFAULT: false



Attribute	Description
AUTOMATIC_RECOVER	The SAPDatabase resource agent tries to recover a failed start attempt automatically one time. This is achieved by performing a forced abort of the RDBMS and/or executing recovery commands. DEFAULT: false
DIR_BOOTSTRAP	The full path to the J2EE instance bootstrap directory. e.g., <i>/usr/sap/COR/JC02/j2ee/cluster/bootstrap</i> This is required only if the DBJ2EE_ONLY parameter is set to true. Note that this directory might be under control of the cluster and not yet mounted at database startup! DEFAULT: <i>/usr/sap/&lt;SID&gt;/*/j2ee/cluster/bootstrap</i>
DIR_SECSTORE	The full path to the J2EE security store directory. This is required only if the DBJ2EE_ONLY parameter is set to true. DEFAULT: <i>/usr/sap/&lt;SID&gt;/SYS/global/security/lib/tools</i>
DB_JARS	The full qualified file name of the jdbc driver for the database connection test. This is required only if the DBJ2EE_ONLY parameter is set to true. It is automatically read from the bootstrap.properties file in Java engine 6.40 and 7.00. For Java engine 7.10, the parameter is mandatory. Example: <i>/oracle/client/10x_64/instantclient/libclntsh.so</i> DEFAULT: empty
PRE_START_USEREXIT, POST_START_USEREXIT, PRE_STOP_USEREXIT, POST_STOP_USEREXIT	Same functionality as in SAPInstance resource agent DEFAULT: empty

**Table 9.4.1.5: Attributes**

See <http://linux-ha.org/doc/man-pages/re-ra-SAPDatabase.html> for the full list of attributes.



## 9.5 Dependencies

### 9.5.1 Resource Dependencies

The resources within a cluster service follow two different dependency rules.

First, the nesting within the service configuration defines startup order and resource dependencies.

In the following example, resource2 depends on resource1. In addition, resource1 is started prior to starting resource2.

```
<service>
  <resource name="resource1">
    <resource name="resource2">
    </resource>
  </resource>
</service>
```

Second, an implicit order and dependency is used. The following lists the implicit startup order of the previously defined resources:

1. lvm
2. fs
3. ip
4. other / application

### 9.5.2 Service Dependencies

The services in a cluster sometimes require dependency rules; e.g., the database must be started prior to starting SAP application servers. However, if the database fails and is subsequently relocated, the application servers should not be restarted.

Additionally, SAP enqueue replication requires a special type of dependency. The enqueue server must always follow the replicated enqueue service.

Note that `central_processing` must be enabled to enable service dependencies and the event scripting required for the follow service dependency:

```
<rm central_processing="1">
  ...
</rm>
```



### 9.5.2.1 Hard and Soft Dependencies

The **rgmanager** service defines inter-service dependencies with soft and hard requirements. A hard dependency would cause the dependent service to be stopped/started if its dependencies were stopped/started. A soft dependency is only valid for initial startup. The dependent service would not be stopped if its dependencies were stopped.

The following example defines the service configuration for a soft dependency:

```
<service name="service1"/>
...
</service>
<service name="service2" depend="service:service1" depend_mode="soft"/>
...
</service>
```

### 9.5.2.2 Follow Service Dependency

The “follow service” dependency makes use of rgmanager's event scripting mechanism. In order to activate the follow service dependency, `central_processing` must be enabled. Also, the following events must be defined in a separate file in the `/usr/share/cluster/` directory on every cluster node. Create and distribute these files according to the (A)SCS and ERS cluster service names.

*/usr/share/cluster/event-service-ers.sl:*

```
notice("Event service triggered!");
evalfile("/usr/share/cluster/follow-service.sl");
follow_service("service:service_ascs", "service:service_ers",
"service:service_ascs");
```

*/usr/share/cluster/event-node-ers.sl:*

```
notice("Event node triggered!");
evalfile("/usr/share/cluster/follow-service.sl");
follow_service("service:service_ascs", "service:service_ers",
"service:service_ascs");
```

The (A)SCS / ERS events can then be embedded in the *cluster.conf* configuration to support enqueue replication with RHEL HA Add-On:

```
<events>
  <event class="service" name="service-ers"
file="/usr/share/cluster/event-service-ers.sl"/>
  <event class="node" name="node-ers" file="/usr/share/cluster/event-
node-ers.sl"/>
</events>
```



# 10 Cluster Management

## 10.1 CMAN

The basic cluster operation can be verified using the **cman\_tool** utility

### 10.1.1 cman\_tool status

The **cman\_tool status** command can be used to show the status of one cluster node:

```
# cman_tool status
Version: 6.2.0
Config Version: 32
Cluster Name: rh_vali
Cluster Id: 13797
Cluster Member: Yes
Cluster Generation: 436
Membership state: Cluster-Member
Nodes: 2
Expected votes: 1
Total votes: 2
Quorum: 1
Active subsystems: 8
Flags: 2node Dirty
Ports Bound: 0 177
Node name: ls3110hb
Node ID: 1
Multicast addresses: 225.0.0.12
Node addresses: 192.168.1.10
```

### 10.1.2 cman\_tool nodes

The **cman\_tool nodes** command can be used to show the status of the basic cluster configuration:

```
# cman_tool nodes
```

Node	Sts	Inc	Joined	Name
1	M	432	2011-07-07 15:15:28	ls3110hb
2	M	436	2011-07-07 15:15:35	ls3121hb



## 10.1.3 cman\_tool services

The **cman\_tool services** command can be used to display the status of the core cluster services:

```
# cman_tool services
type          level name      id        state
fence         0      default  00010001 none
[1 2]
dlm           1      rgmanager 00020001 none
[1 2]
```

## 10.2 rgmanager

### 10.2.1 clustat

The resource group or service manager state and all configured services can be displayed with the **clustat** command:

```
# clustat -l
Cluster Status for rh_vali @ Tue Jul 12 11:10:16 2011
Member Status: Quorate

Member Name                                ID    Status
-----
ls3110hb                                  1 Online, Local, RG-Master
ls3121hb                                  2 Online, RG-Worker

Service Information
-----

Service Name      : service:svc_cor_database
Current State     : started (112)
Flags             : none (0)
Owner             : ls3121hb
Last Owner        : ls3110hb
Last Transition   : Fri Jul  8 12:23:34 2011
```





```
Service Name      : service:svc_cor_ers01
Current State     : started (112)
Flags             : none (0)
Owner            : ls3121hb
Last Owner       : ls3110hb
Last Transition   : Fri Jul  8 10:38:02 2011
```

```
Service Name      : service:svc_cor_d02
Current State     : started (112)
Flags             : none (0)
Owner            : ls3110hb
Last Owner       : ls3121hb
Last Transition   : Fri Jul  8 11:02:18 2011
```

```
Service Name      : service:svc_cor_scs00
Current State     : started (112)
Flags             : none (0)
Owner            : ls3110hb
Last Owner       : ls3110hb
Last Transition   : Fri Jul  8 10:16:53 2011
```

## 10.2.2 clusvcadm

The resource manager services can be controlled by the **clusvcadm** command. The basic operations are:

```
clusvcadm -e <service> -F
```

starts service <service> according to failover domain rules

```
clusvcadm -r <service> -m <member>
```

relocate service <service> to member <member>

```
clusvcadm -d <service>
```

disables/stops service <service>

For detailed information, reference the clusvcadm(8) manpage.



### 10.2.3 rg\_test

The resource manager cluster configuration can be verified using the **rg\_test** command:

```
# rg_test test /etc/cluster/cluster.conf
```

```
Running in test mode.
```

```
Loaded 22 resource rules
```

```
=== Resources List ===
```

```
Resource type: ip
```

```
Instances: 1/1
```

```
Agent: ip.sh
```

```
Attributes:
```

```
[ ... output abbreviated ... ]
```

```
=== Failover Domains ===
```

```
Failover domain: ls3110
```

```
Flags: Ordered Restricted
```

```
Node ls3110hb (id 1, priority 1)
```

```
Node ls3121hb (id 2, priority 2)
```

```
Failover domain: ls3121
```

```
Flags: Ordered Restricted
```

```
Node ls3121hb (id 2, priority 1)
```

```
Node ls3110hb (id 1, priority 2)
```

```
[ ... output truncated ... ]
```



# 11 Closing thoughts

- While planning a cluster consider everything from hardware to software and prepare answers to the following questions:
  - Is there no single-point-of-failure in the hardware setup?
  - Is the SAN configuration disaster tolerant and performance optimized (preferred path / ALUA, check with the appropriate storage vendor to use the right FC kernel module, HBA FirmWare etc.)
  - Does the network infrastructure support hardware bonding?
    - IEEE 802.3ad Dynamic Link Aggregation (LACP) is a standard that makes switches and other network devices aware of port trunking / bonding. If the network switches support IEEE 802.3ad, bonded ports for LACP can be configured on the switch and bonding “mode=4” in the bond definition on the linux operating system level.
    - If using LACP is not possible try a soft-bond mode (balancing, active-backup, etc.) that doesn't require the network switches to be bond-aware
  - Does the backup solution still work in the clustered environment?
- Conduct thorough testing of all failover / split-brain scenarios
  - It is very important to verify the continued availability by simulating:
    - power outages (pull the plugs! Don't just use the soft-power-off)
    - network failures (un-plug the cables)
    - SAN failures (depending on configuration)
  - Consider that the fencing devices (network-based, remote console based) might not work during a power outage however Service-failovers depend on a successful fence attempt



# Appendix A: Cluster Configuration Files

The following is an example `/etc/cluster/cluster.conf` file for a 2-node cluster with an Oracle database instance, SAP Central Services instance (SCS), Enqueue replication service (ERS) and a SAP Primary Application Server (PAS) instance.

Two failover domains were defined in order to choose the preferred executing node for each service during normal operation.

*/etc/cluster/cluster.conf:*

```
<?xml version="1.0"?>
<cluster alias="rh_vali" config_version="32" name="rh_vali">
  <fence_daemon post_fail_delay="2" post_join_delay="3"/>
  <clusternodes>
    <clusternode name="ls3110hb" nodeid="1" votes="1">
      <fence>
        <method name="1">
          <device name="ls3110r"/>
        </method>
        <method name="2">
          <device name="manual_fence"
nodename="ls3110hb"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="ls3121hb" nodeid="2" votes="1">
      <fence>
        <method name="1">
          <device name="ls3121r"/>
        </method>
        <method name="2">
          <device name="manual_fence"
nodename="ls3121hb"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <cman expected_votes="1" two_node="1">
    <multicast addr="225.0.0.12"/>
  </cman>
```



```
<fencedevices>
  <fencedevice agent="fence_rsa" ipaddr="10.20.88.227"
login="USERID" name="ls3110r" passwd="PASSWORD"/>
  <fencedevice agent="fence_rsa" ipaddr="10.20.88.229"
login="USERID" name="ls3121r" passwd="PASSWORD"/>
  <fencedevice agent="fence_manual" name="manual_fence"/>
</fencedevices>
<rm central_processing="1">
  <failoverdomains>
    <failoverdomain name="ls3110" ordered="1" restricted="1">
      <failoverdomainnode name="ls3110hb" priority="1"/>
      <failoverdomainnode name="ls3121hb" priority="2"/>
    </failoverdomain>
    <failoverdomain name="ls3121" ordered="1" restricted="1">
      <failoverdomainnode name="ls3121hb" priority="1"/>
      <failoverdomainnode name="ls3110hb" priority="2"/>
    </failoverdomain>
  </failoverdomains>
  <resources>
    <ip address="192.168.1.11/24" monitor_link="1"/>
    <ip address="192.168.1.12/24" monitor_link="1"/>
    <ip address="192.168.1.13/24" monitor_link="1"/>
    <ip address="192.168.1.14/24" monitor_link="1"/>
    <lvm lv_name="lv_cor_scs" name="res_scs_LVM"
vg_name="vg_cor_scs"/>
    <lvm lv_name="lv_cor_ers" name="res_ers_LVM"
vg_name="vg_cor_ers"/>
    <lvm lv_name="lv_cor_db" name="res_db_LVM"
vg_name="vg_cor_db"/>
    <lvm lv_name="lv_cor_d02" name="res_D02_LVM"
vg_name="vg_cor_d02"/>
    <fs device="/dev/mapper/vg_cor_db-lv_cor_db" force_fsck="0"
force_unmount="1" fsid="20211" fstype="ext3" mountpoint="/oracle/COR"
name="res_db_FS" options="" self_fence="0"/>
    <fs device="/dev/mapper/vg_cor_scs-lv_cor_scs"
force_fsck="0" force_unmount="1" fsid="2773" fstype="ext3"
mountpoint="/usr/sap/COR/SCS00" name="res_scs_FS" options=""
self_fence="0"/>
    <fs device="/dev/mapper/vg_cor_d02-lv_cor_d02"
force_fsck="0" force_unmount="1" fsid="16525" fstype="ext3"
mountpoint="/usr/sap/COR/D02" name="res_D02_FS" options="" self_fence="0"/>
```



```

        <fs device="/dev/mapper/vg_cor_sap-lv_cor_sap"
force_fsck="0" force_unmount="1" fsid="26023" fstype="ext3"
mountpoint="/usr/sap/COR" name="res_sap_wa_FS" options="" self_fence="0"/>

        <fs device="/dev/mapper/vg_cor_ers-lv_cor_ers"
force_fsck="0" force_unmount="1" fsid="21857" fstype="ext3"
mountpoint="/usr/sap/COR/ERS01" name="res_ers_FS" options=""
self_fence="0"/>

        <SAPDatabase AUTOMATIC_RECOVER="TRUE" DBJ2EE_ONLY=""
DBTYPE="ORA" DB_JARS="" DIR_BOOTSTRAP="" DIR_EXECUTABLE="/sapmnt/COR/exe"
DIR_SECSTORE="" JAVA_HOME="" NETSERVICENAME="" POST_START_USEREXIT=""
POST_STOP_USEREXIT="" PRE_START_USEREXIT="" PRE_STOP_USEREXIT="" SID="COR"
STRICT_MONITORING="FALSE"/>

        <SAPInstance AUTOMATIC_RECOVER="TRUE"
DIR_EXECUTABLE="/sapmnt/COR/exe" DIR_PROFILE="/sapmnt/COR/profile"
InstanceName="COR_SCS00_cor-scs" POST_START_USEREXIT=""
POST_STOP_USEREXIT="" PRE_START_USEREXIT="" PRE_STOP_USEREXIT=""
START_PROFILE="/sapmnt/COR/profile/START_SCS00_cor-scs" START_WAITTIME=""/>

        <SAPInstance AUTOMATIC_RECOVER="TRUE"
DIR_EXECUTABLE="/sapmnt/COR/exe" DIR_PROFILE="/sapmnt/COR/profile"
InstanceName="COR_D02_cor-pas" POST_START_USEREXIT="" POST_STOP_USEREXIT=""
PRE_START_USEREXIT="" PRE_STOP_USEREXIT=""
START_PROFILE="/sapmnt/COR/profile/START_D02_cor-pas" START_WAITTIME=""/>

        <SAPInstance AUTOMATIC_RECOVER="TRUE"
DIR_EXECUTABLE="/sapmnt/COR/exe" DIR_PROFILE="/sapmnt/COR/profile"
InstanceName="COR_ERS01_cor-ers" POST_START_USEREXIT=""
POST_STOP_USEREXIT="" PRE_START_USEREXIT="" PRE_STOP_USEREXIT=""
START_PROFILE="/sapmnt/COR/profile/START_ERS01_cor-ers" START_WAITTIME=""/>

    </resources>

    <service autostart="1" domain="ls3121" name="svc_cor_database"
recovery="relocate">

        <ip ref="192.168.1.12/24"/>

        <lvm ref="res_db_LVM"/>

        <fs ref="res_db_FS"/>

        <SAPDatabase ref="COR"/>

    </service>

    <service autostart="1" domain="ls3110" name="svc_cor_scs00"
recovery="relocate">

        <ip ref="192.168.1.11/24"/>

        <lvm ref="res_scs_LVM"/>

        <fs ref="res_scs_FS"/>

        <SAPInstance ref="COR_SCS00_cor-scs"/>

    </service>

    <service autostart="1" depend="service:svc_cor_scs00"
depend_mode="soft" domain="ls3110" name="svc_cor_d02" recovery="relocate">

        <ip ref="192.168.1.13/24"/>

```



```

        <lvm ref="res_D02_LVM"/>
        <fs ref="res_D02_FS"/>
        <SAPInstance ref="COR_D02_cor-pas"/>
    </service>
    <service autostart="1" domain="ls3121" name="svc_cor_ers01"
recovery="relocate">
        <ip ref="192.168.1.14/24"/>
        <lvm ref="res_ers_LVM"/>
        <fs ref="res_ers_FS"/>
        <SAPInstance ref="COR_ERS01_cor-ers"/>
    </service>
    <events>
        <event class="service" name="service-ers"
file="/usr/share/cluster/event-service-ers.sl"/>
        <event class="node" name="node-ers"
file="/usr/share/cluster/event-node-ers.sl"/>
    </events>
</rm>
</cluster>

```

In addition to the cluster.conf these two files referenced by the event tag were created on every cluster node:

*/usr/share/cluster/event-service-ers.sl:*

```

notice("Event service triggered!");
evalfile("/usr/share/cluster/follow-service.sl");
follow_service("service:svc_cor_scs00", "service:svc_cor_ers01",
"service:svc_cor_scs00");

```

*/usr/share/cluster/event-node-ers.sl:*

```

notice("Event node triggered!");
evalfile("/usr/share/cluster/follow-service.sl");
follow_service("service:svc_cor_scs00", "service:svc_cor_ers01",
"service:svc_cor_scs00");

```



# Appendix B: Reference Documentation

The following list includes the existing documentation and articles referenced by this document.

1. **Red Hat Enterprise Linux Release Notes and Installation Guide**  
[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/index.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/index.html)
2. **Configuring and Managing a Red Hat Cluster**  
[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5/html/Cluster\\_Administration/index.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Cluster_Administration/index.html)
3. **SAP Installation Guides**  
<http://service.sap.com/instguides>
4. **SAP Technical Infrastructure Guide (high-availability)**  
<https://www.sdn.sap.com/irj/sdn/ha>
5. **How to Optimally Configure a Quorum Disk in Red Hat Enterprise Linux Clustering and High-Availability Environments**  
<https://access.redhat.com/knowledge/techbriefs/how-optimally-configure-quorum-disk-red-hat-enterprise-linux-clustering-and-hig>
6. **Event Scripting**  
<https://fedorahosted.org/cluster/wiki/EventScripting>
7. **Red Hat Enterprise Linux Cluster, High Availability Knowledge Base Index**  
<https://access.redhat.com/kb/docs/DOC-48718>
8. **Support for Red Hat Enterprise Linux Cluster and High Availability Stretch Architectures**  
<https://access.redhat.com/kb/docs/DOC-58412>
9. **Architecture Review Process for Red Hat Enterprise Linux High Availability, Clustering, and GFS/GFS2**  
<https://access.redhat.com/kb/docs/DOC-53348>
10. **How to Control Access to Shared Storage Devices Using SCSI Persistent Reservations with Red Hat Enterprise Linux Clustering and High Availability**  
<https://access.redhat.com/knowledge/techbriefs/how-control-access-shared-storage-devices-using-scsi-persistent-reservations-re>
11. **Red Hat Logical Volume Manager Administration Guide**  
[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5/html/Logical\\_Volume\\_Manager\\_Administration/index.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Logical_Volume_Manager_Administration/index.html)





## 12. Cluster Administration Guide

[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/5/html/Cluster\\_Administration/index.html](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Cluster_Administration/index.html)



## Appendix C: Revision History

Revision 2.0	Monday August 15, 2011	Alfredo Moralejo
Added Stretch Cluster contents – Alfredo Moralejo		
Formatting and layout by John Herr		
Revision 1.0	Monday May 23, 2011	Martin Tegtmeier, Realtech Frank Danapfel, Software Engineer (on-site at SAP)
Initial Release		
Formatting and layout by John Herr		