Red Hat Cluster Suite

Configuring and Managing a Cluster

Red Hat Cluster Suite: Configuring and Managing a Cluster

Red Hat, Inc.

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

rh-cs(EN)-4-Print-RHI (2006-03-07T17:50)

The GPG fingerprint of the 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

Table of Contents

Introduction

The Red Hat Cluster Suite is a collection of technologies working together to provide data integrity and the ability to maintain application availability in the event of a failure. Administrators can deploy enterprise cluster solutions using a combination of hardware redundancy along with the failover and load-balancing technologies in Red Hat Cluster Suite.

Red Hat Cluster Manager is a high-availability cluster solution specifically suited for database applications, network file servers, and World Wide Web (Web) servers with dynamic content. A Red Hat Cluster Manager system features data integrity and application availability using redundant hardware, shared disk storage, power management, robust cluster communication, and robust application failover mechanisms.

Administrators can also deploy highly available applications services using Piranha, a load-balancing and advanced routing cluster solution based on Linux Virtual Server (LVS) technology. Using Piranha, administrators can build highly available e-commerce sites that feature complete data integrity and service availability, in addition to load balancing capabilities. Refer to Part II *Configuring a Linux Virtual Server Cluster* for more information.

This guide assumes that the user has an advanced working knowledge of Red Hat Enterprise Linux and understands the concepts of server computing. For more information about using Red Hat Enterprise Linux, refer to the following resources:

- *Red Hat Enterprise Linux Installation Guide* for information regarding installation.
- *Red Hat Enterprise Linux Introduction to System Administration* for introductory information for new Red Hat Enterprise Linux system administrators.
- *Red Hat Enterprise Linux System Administration Guide* for more detailed information about configuring Red Hat Enterprise Linux to suit your particular needs as a user.
- *Red Hat Enterprise Linux Reference Guide* provides detailed information suited for more experienced users to reference when needed, as opposed to step-by-step instructions.
- *Red Hat Enterprise Linux Security Guide* details the planning and the tools involved in creating a secured computing environment for the data center, workplace, and home.

HTML, PDF, and RPM versions of the manuals are available on the Red Hat Enterprise Linux Documentation CD and online at:

`http://www.redhat.com/docs/`

1. How To Use This Manual

This manual contains information about setting up a Red Hat Cluster Manager system. These tasks are described in the following chapters:

- Chapter 2 *Hardware Installation and Operating System Configuration*
- Chapter 3 *Installing and Configuring Red Hat Cluster Suite Software*

Part II *Configuring a Linux Virtual Server Cluster* describes how to achieve load balancing in an Red Hat Enterprise Linux cluster by using the Linux Virtual Server.

Appendix A *Supplementary Hardware Information* contains detailed configuration information on specific hardware devices and shared storage configurations.

Appendix B *Selectively Installing Red Hat Cluster Suite Packages* contains information about custom installation of Red Hat Cluster Suite and Red Hat GFS RPMs.

Appendix C *`Multipath-usage.txt` File for Red Hat Enterprise Linux 4 Update 3* contains information from the `Multipath-usage.txt` file. The file provides guidelines for using `dm-multipath` with Red Hat Cluster Suite for Red Hat Enterprise Linux 4 Update 3.

This guide assumes you have a thorough understanding of Red Hat Enterprise Linux system administration concepts and tasks. For detailed information on Red Hat Enterprise Linux system administration, refer to the *Red Hat Enterprise Linux System Administration Guide*. For reference information on Red Hat Enterprise Linux, refer to the *Red Hat Enterprise Linux Reference Guide*.

2. Document Conventions

In this manual, certain words are represented in different fonts, typefaces, sizes, and weights. This highlighting is systematic; different words are represented in the same style to indicate their inclusion in a specific category. The types of words that are represented this way include the following:

`command`

> Linux commands (and other operating system commands, when used) are represented this way. This style should indicate to you that you can type the word or phrase on the command line and press [Enter] to invoke a command. Sometimes a command contains words that would be displayed in a different style on their own (such as file names). In these cases, they are considered to be part of the command, so the entire phrase is displayed as a command. For example:
>
> Use the `cat testfile` command to view the contents of a file, named `testfile`, in the current working directory.

`file name`

> File names, directory names, paths, and RPM package names are represented this way. This style indicates that a particular file or directory exists with that name on your system. Examples:

The `.bashrc` file in your home directory contains bash shell definitions and aliases for your own use.

The `/etc/fstab` file contains information about different system devices and file systems.

Install the `webalizer` RPM if you want to use a Web server log file analysis program.

application

This style indicates that the program is an end-user application (as opposed to system software). For example:

Use **Mozilla** to browse the Web.

[key]

A key on the keyboard is shown in this style. For example:

To use [Tab] completion, type in a character and then press the [Tab] key. Your terminal displays the list of files in the directory that start with that letter.

[key]-[combination]

A combination of keystrokes is represented in this way. For example:

The [Ctrl]-[Alt]-[Backspace] key combination exits your graphical session and returns you to the graphical login screen or the console.

text found on a GUI interface

A title, word, or phrase found on a GUI interface screen or window is shown in this style. Text shown in this style indicates that a particular GUI screen or an element on a GUI screen (such as text associated with a checkbox or field). Example:

Select the **Require Password** checkbox if you would like your screensaver to require a password before stopping.

top level of a menu on a GUI screen or window

A word in this style indicates that the word is the top level of a pulldown menu. If you click on the word on the GUI screen, the rest of the menu should appear. For example:

Under **File** on a GNOME terminal, the **New Tab** option allows you to open multiple shell prompts in the same window.

Instructions to type in a sequence of commands from a GUI menu look like the following example:

Go to **Applications** (the main menu on the panel) => **Programming => Emacs Text Editor** to start the **Emacs** text editor.

button on a GUI screen or window

This style indicates that the text can be found on a clickable button on a GUI screen. For example:

Click on the **Back** button to return to the webpage you last viewed.

`computer output`

Text in this style indicates text displayed to a shell prompt such as error messages and responses to commands. For example:

The `ls` command displays the contents of a directory. For example:

```
Desktop                 about.html       logs           paulwesterberg.png
Mail                    backupfiles      mail           reports
```

The output returned in response to the command (in this case, the contents of the directory) is shown in this style.

`prompt`

A prompt, which is a computer's way of signifying that it is ready for you to input something, is shown in this style. Examples:

```
$
```

```
#
```

```
[stephen@maturin stephen]$
```

```
leopard login:
```

`user input`

Text that the user types, either on the command line or into a text box on a GUI screen, is displayed in this style. In the following example, **`text`** is displayed in this style:

To boot your system into the text based installation program, you must type in the **`text`** command at the `boot:` prompt.

<replaceable>

Text used in examples that is meant to be replaced with data provided by the user is displayed in this style. In the following example, *<version-number>* is displayed in this style:

The directory for the kernel source is `/usr/src/kernels/`*<version-number>*`/`, where *<version-number>* is the version and type of kernel installed on this system.

Additionally, we use several different strategies to draw your attention to certain pieces of information. In order of urgency, these items are marked as a note, tip, important, caution, or warning. For example:

Note

Remember that Linux is case sensitive. In other words, a rose is not a ROSE is not a rOsE.

Tip

The directory `/usr/share/doc/` contains additional documentation for packages installed on your system.

Important

If you modify the DHCP configuration file, the changes do not take effect until you restart the DHCP daemon.

Caution

Do not perform routine tasks as root — use a regular user account unless you need to use the root account for system administration tasks.

Warning

Be careful to remove only the necessary partitions. Removing other partitions could result in data loss or a corrupted system environment.

3. More to Come

This manual is part of Red Hat's growing commitment to provide useful and timely support to Red Hat Enterprise Linux users.

3.1. Send in Your Feedback

If you spot a typo, or if you have thought of a way to make this manual better, we would love to hear from you. Please submit a report in Bugzilla (http://bugzilla.redhat.com/bugzilla/) against the component `rh-cs`.

Be sure to mention the manual's identifier:

```
rh-cs(EN)-4-Print-RHI (2006-03-07T17:50)
```

By mentioning this manual's identifier, we know exactly which version of the guide you have.

If you have a suggestion for improving the documentation, try to be as specific as possible. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

4. Activate Your Subscription

Before you can access service and software maintenance information, and the support documentation included in your subscription, you must activate your subscription by registering with Red Hat. Registration includes these simple steps:

- Provide a Red Hat login
- Provide a subscription number
- Connect your system

The first time you boot your installation of Red Hat Enterprise Linux, you are prompted to register with Red Hat using the **Setup Agent**. If you follow the prompts during the **Setup Agent**, you can complete the registration steps and activate your subscription.

If you can not complete registration during the **Setup Agent** (which requires network access), you can alternatively complete the Red Hat registration process online at http://www.redhat.com/register/.

4.1. Provide a Red Hat Login

If you do not have an existing Red Hat login, you can create one when prompted during the **Setup Agent** or online at:

```
https://www.redhat.com/apps/activate/newlogin.html
```

A Red Hat login enables your access to:

- Software updates, errata and maintenance via Red Hat Network
- Red Hat technical support resources, documentation, and Knowledgebase

If you have forgotten your Red Hat login, you can search for your Red Hat login online at:

```
https://rhn.redhat.com/help/forgot_password.pxt
```

4.2. Provide Your Subscription Number

Your subscription number is located in the package that came with your order. If your package did not include a subscription number, your subscription was activated for you and you can skip this step.

You can provide your subscription number when prompted during the **Setup Agent** or by visiting http://www.redhat.com/register/.

4.3. Connect Your System

The Red Hat Network Registration Client helps you connect your system so that you can begin to get updates and perform systems management. There are three ways to connect:

1. During the **Setup Agent** — Check the **Send hardware information** and **Send system package list** options when prompted.
2. After the **Setup Agent** has been completed — From **Applications** (the main menu on the panel), go to **System Tools**, then select **Red Hat Network**.
3. After the **Setup Agent** has been completed — Enter the following command from the command line as the root user:
   - `/usr/bin/up2date --register`

I. Using the Red Hat Cluster Manager

Clustered systems provide reliability, scalability, and availability to critical production services. Using the Red Hat Cluster Manager, administrators can create high availability clusters for filesharing, Web servers, and more. This part discusses the installation and configuration of cluster systems using the recommended hardware and Red Hat Enterprise Linux.

Table of Contents

Chapter 1.

Red Hat Cluster Manager Overview

Red Hat Cluster Manager allows administrators to connect separate systems (called *members* or *nodes*) together to create failover clusters that ensure application availability and data integrity under several failure conditions. Administrators can use Red Hat Cluster Manager with database applications, file sharing services, web servers, and more.

To set up a failover cluster, you must connect the nodes to the cluster hardware, and configure the nodes into the cluster environment. The foundation of a cluster is an advanced host membership algorithm. This algorithm ensures that the cluster maintains complete data integrity by using the following methods of inter-node communication:

- Network connections between the cluster systems
- A Cluster Configuration System daemon (ccsd) that synchronizes configuration between cluster nodes

To make an application and data highly available in a cluster, you must configure a *cluster service*, an application that would benefit from Red Hat Cluster Manager to ensure high availability. A cluster service is made up of cluster *resources*, components that can be failed over from one node to another, such as an IP address, an application initialization script, or a Red Hat GFS shared partition. Building a cluster using Red Hat Cluster Manager allows transparent client access to cluster services. For example, you can provide clients with access to highly-available database applications by building a cluster service using Red Hat Cluster Manager to manage service availability and shared Red Hat GFS storage partitions for the database data and end-user applications.

You can associate a cluster service with a *failover domain*, a subset of cluster nodes that are eligible to run a particular cluster service. In general, any eligible, properly-configured node can run the cluster service. However, each cluster service can run on only one cluster node at a time in order to maintain data integrity. You can specify whether or not the nodes in a failover domain are ordered by preference. You can also specify whether or not a cluster service is restricted to run only on nodes of its associated failover domain. (When associated with an unrestricted failover domain, a cluster service can be started on any cluster node in the event no member of the failover domain is available.)

You can set up an *active-active* configuration in which the members run different cluster services simultaneously, or a *hot-standby* configuration in which primary members run all the cluster services, and a backup member takes over only if a primary member fails.

If a hardware or software failure occurs, the cluster automatically restarts the failed node's cluster services on the functional node. This *cluster-service failover* capability ensures that no data is lost, and there is little disruption to users. When the failed node recovers, the cluster can re-balance the cluster services across the nodes.

In addition, you can cleanly stop the cluster services running on a cluster system and then restart them on another system. This *cluster-service relocation* capability allows you to maintain application and data availability when a cluster node requires maintenance.

1.1. Red Hat Cluster Manager Features

Cluster systems deployed with Red Hat Cluster Manager include the following features:

No-single-point-of-failure hardware configuration

> Clusters can include a dual-controller RAID array, multiple bonded network channels, multiple paths between cluster members and storage, and redundant uninterruptible power supply (UPS) systems to ensure that no single failure results in application down time or loss of data.
>
> **Note**
>
> For information about using `dm-multipath` with Red Hat Cluster Suite, refer toAppendix C *`Multipath-usage.txt` File for Red Hat Enterprise Linux 4 Update 3*
>
> Alternatively, a low-cost cluster can be set up to provide less availability than a no-single-point-of-failure cluster. For example, you can set up a cluster with a single-controller RAID array and only a single Ethernet channel.
>
> Certain low-cost alternatives, such as host RAID controllers, software RAID without cluster support, and multi-initiator parallel SCSI configurations are not compatible or appropriate for use as shared cluster storage.

Cluster configuration and administration framework

> Red Hat Cluster Manager allows you to easily configure and administer cluster services to make resources such as applications, server daemons, and shared data highly available. To create a cluster service, you specify the resources used in the cluster service as well as the properties of the cluster service, such as the cluster service name, application initialization (init) scripts, disk partitions, mount points, and the cluster nodes on which you prefer the cluster service to run. After you add a cluster service, the cluster management software stores the information in a cluster configuration file, and the configuration data is aggregated to all cluster nodes using the *Cluster Configuration System* (or CCS), a daemon installed on each cluster node that allows retrieval of changes to the XML-based `/etc/cluster/cluster.conf` configuration file.
>
> Red Hat Cluster Manager provides an easy-to-use framework for database applications. For example, a database cluster service serves highly-available data to a database application. The application running on a cluster node provides network access to database client systems, such as Web applications. If the cluster service fails over to another node, the application can still access the shared database data. A

network-accessible database cluster service is usually assigned an IP address, which is failed over along with the cluster service to maintain transparent access for clients.

The cluster-service framework can also easily extend to other applications through the use of customized init scripts.

Cluster administration user interface

The Red Hat Cluster Suite management graphical user interface (GUI) facilitates the administration and monitoring tasks of cluster resources such as the following: creating, starting, and stopping cluster services; relocating cluster services from one node to another; modifying the cluster service configuration; and monitoring the cluster nodes. The CMAN interface allows administrators to individually control the cluster on a per-node basis.

Failover domains

By assigning a cluster service to a *restricted failover domain*, you can limit the nodes that are eligible to run a cluster service in the event of a failover. (A cluster service that is assigned to a restricted failover domain cannot be started on a cluster node that is not included in that failover domain.) You can order the nodes in a failover domain by preference to ensure that a particular node runs the cluster service (as long as that node is active). If a cluster service is assigned to an unrestricted failover domain, the cluster service starts on any available cluster node (if none of the nodes of the failover domain are available).

Data integrity assurance

To ensure data integrity, only one node can run a cluster service and access cluster-service data at one time. The use of power switches in the cluster hardware configuration enables a node to power-cycle another node before restarting that node's cluster services during the failover process. This prevents any two systems from simultaneously accessing the same data and corrupting it. It is strongly recommended that *fence devices* (hardware or software solutions that remotely power, shutdown, and reboot cluster nodes) are used to guarantee data integrity under all failure conditions. Watchdog timers are an alternative used to ensure correct operation of cluster service failover.

Ethernet channel bonding

To monitor the health of the other nodes, each node monitors the health of the remote power switch, if any, and issues heartbeat pings over network channels. With Ethernet channel bonding, multiple Ethernet interfaces are configured to behave as one, reducing the risk of a single-point-of-failure in the typical switched Ethernet connection between systems.

Cluster-service failover capability

If a hardware or software failure occurs, the cluster takes the appropriate action to maintain application availability and data integrity. For example, if a node completely fails, a healthy node (in the associated failover domain, if used) starts the service or services that the failed node was running prior to failure. Cluster services already running on the healthy node are not significantly disrupted during the failover process.

Note

For Red Hat Cluster Suite 4, node health is monitored through a cluster network heartbeat. In previous versions of Red Hat Cluster Suite, node health was monitored on shared disk. Shared disk is *not* required for node-health monitoring in Red Hat Cluster Suite 4.

When a failed node reboots, it can rejoin the cluster and resume running the cluster service. Depending on how the cluster services are configured, the cluster can rebalance services among the nodes.

Manual cluster-service relocation capability

In addition to automatic cluster-service failover, a cluster allows you to cleanly stop cluster services on one node and restart them on another node. You can perform planned maintenance on a node system while continuing to provide application and data availability.

Event logging facility

To ensure that problems are detected and resolved before they affect cluster-service availability, the cluster daemons log messages by using the conventional Linux syslog subsystem.

Application monitoring

The infrastructure in a cluster monitors the state and health of an application. In this manner, should an application-specific failure occur, the cluster automatically restarts the application. In response to the application failure, the application attempts to be restarted on the node it was initially running on; failing that, it restarts on another cluster node. You can specify which nodes are eligible to run a cluster service by assigning a failover domain to the cluster service.

1.1.1. Red Hat Cluster Manager Subsystem Overview

Table 1-1 summarizes the GFS Software subsystems and their components.

Software Subsystem	Components	Description
Cluster Configuration Tool	`system-config-cluster`	Command used to manage cluster configuration in a graphical setting.
Cluster Configuration System (CCS)	`ccs_tool`	Notifies `ccsd` of an updated `cluster.conf` file. Also, used for upgrading a configuration file from a Red Hat GFS 6.0 (or earlier) cluster to the format of the Red Hat Cluster Suite 4 configuration file.
	`ccs_test`	Diagnostic and testing command that is used to retrieve information from configuration files through `ccsd`.
	`ccsd`	CCS daemon that runs on all cluster nodes and provides configuration file data to cluster software.
Resource Group Manager (rgmanager)	`clusvcadm`	Command used to manually enable, disable, relocate, and restart user services in a cluster
	`clustat`	Command used to display the status of the cluster, including node membership and services running.
	`clurgmgrd`	Daemon used to handle user service requests including service start, service disable, service relocate, and service restart
Fence	`fence_ack_manual`	User interface for `fence_manual` agent.
	`fence_apc`	Fence agent for APC power switch.
	`fence_bladecenter`	Fence agent for for IBM Bladecenters with Telnet interface.
	`fence_brocade`	Fence agent for Brocade Fibre Channel switch.

Software Subsystem	Components	Description
	`fence_bullpap`	Fence agent for Bull Novascale Platform Administration Processor (PAP) Interface.
	`fence_drac`	Fence agent for Dell Remote Access Controller/Modular Chassis (DRAC/MC).
	`fence_egenera`	Fence agent used with Egenera BladeFrame system.
	`fence_gnbd`	Fence agent used with GNBD storage.
	`fence_ilo`	Fence agent for HP ILO interfaces (formerly fence_rib).
	`fence_ipmilan`	Fence agent for Intelligent Platform Management Interface (IPMI).
	`fence_manual`	Fence agent for manual interaction. Note: Manual fencing is *not* supported for production environments.
	`fence_mcdata`	Fence agent for McData Fibre Channel switch.
	`fence_node`	Command used by `lock_gulmd` when a fence operation is required. This command takes the name of a node and fences it based on the node's fencing configuration.
	`fence_rps10`	Fence agent for WTI Remote Power Switch, Model RPS-10 (Only used with two-node clusters).
	`fence_rsa`	Fence agent for IBM Remote Supervisor Adapter II (RSA II).
	`fence_sanbox2`	Fence agent for SANBox2 Fibre Channel switch.
	`fence_vixel`	Fence agent for Vixel Fibre Channel switch.

Software Subsystem	Components	Description
	`fence_wti`	Fence agent for WTI power switch.
	`fenced`	The fence daemon. Manages the fence domain.
DLM	`libdlm.so.1.0.0`	Library for Distributed Lock Manager (DLM) support.
	`dlm.ko`	Kernel module that is installed on cluster nodes for Distributed Lock Manager (DLM) support.
LOCK_GULM	`lock_gulm.o`	Kernel module that is installed on GFS nodes using the LOCK_GULM lock module.
	`lock_gulmd`	Server/daemon that runs on each node and communicates with all nodes in GFS cluster.
	`libgulm.so.`*`xxx`*	Library for GULM lock manager support
	`gulm_tool`	Command that configures and debugs the `lock_gulmd` server.
LOCK_NOLOCK	`lock_nolock.o`	Kernel module installed on a node using GFS as a local file system.
GNBD	`gnbd.o`	Kernel module that implements the GNBD device driver on clients.
	`gnbd_serv.o`	Kernel module that implements the GNBD server. It allows a node to export local storage over the network.
	`gnbd_export`	Command to create, export and manage GNBDs on a GNBD server.
	`gnbd_import`	Command to import and manage GNBDs on a GNBD client.

Table 1-1. Red Hat Cluster Manager Software Subsystem Components

Chapter 2.

Hardware Installation and Operating System Configuration

To set up the hardware configuration and install Red Hat Enterprise Linux, follow these steps:

- Choose a cluster hardware configuration that meets the needs of applications and users; refer to Section 2.1 *Choosing a Hardware Configuration*.
- Set up and connect the members and the optional console switch and network switch or hub; refer to Section 2.3 *Setting Up the Nodes*.
- Install and configure Red Hat Enterprise Linux on the cluster members; refer to Section 2.4 *Installing and Configuring Red Hat Enterprise Linux*.
- Set up the remaining cluster hardware components and connect them to the members; refer to Section 2.5 *Setting Up and Connecting the Cluster Hardware*.

After setting up the hardware configuration and installing Red Hat Enterprise Linux, install the cluster software.

2.1. Choosing a Hardware Configuration

The Red Hat Cluster Manager allows administrators to use commodity hardware to set up a cluster configuration that meets the performance, availability, and data integrity needs of applications and users. Cluster hardware ranges from low-cost minimum configurations that include only the components required for cluster operation, to high-end configurations that include redundant Ethernet channels, hardware RAID, and power switches.

Regardless of configuration, the use of high-quality hardware in a cluster is recommended, as hardware malfunction is a primary cause of system down time.

Although all cluster configurations provide availability, some configurations protect against every *single point of failure*. In addition, all cluster configurations provide data integrity, but some configurations protect data under every failure condition. Therefore, administrators must fully understand the needs of their computing environment and also the availability and data integrity features of different hardware configurations to choose the cluster hardware that meets the requirements.

When choosing a cluster hardware configuration, consider the following:

Performance requirements of applications and users

> Choose a hardware configuration that provides adequate memory, CPU, and I/O resources. Be sure that the configuration chosen can handle any future increases in workload as well.

Cost restrictions

> The hardware configuration chosen must meet budget requirements. For example, systems with multiple I/O ports usually cost more than low-end systems with fewer expansion capabilities.

Availability requirements

> In a mission-critical production environment, a cluster hardware configuration must protect against all single points of failure, including: disk, storage interconnect, Ethernet channel, and power failure. Environments that can tolerate an interruption in availability (such as development environments) may not require as much protection.

Data integrity under all failure conditions requirement

> Using fence devices in a cluster configuration ensures that service data is protected under every failure condition. These devices enable a node to power cycle another node before restarting its services during failover. Power switches protect against data corruption in cases where an unresponsive (or hung) node tries to write data to the disk after its replacement node has taken over its services.
>
> If you are not using power switches in the cluster, cluster service failures can result in services being run on more than one node, which can cause data corruption. Refer to Section 2.5.2 *Configuring a Fence Device* for more information about the benefits of using power switches in a cluster. It is required that production environments use power switches in the cluster hardware configuration.

2.1.1. Minimum Hardware Requirements

A *minimum hardware configuration* includes only the hardware components that are required for cluster operation, as follows:

- At least two servers to run cluster services
- Ethernet connection for sending heartbeat pings and for client network access
- Network switch or hub to connect cluster nodes and resources
- A fence device

The hardware components described in Table 2-1 can be used to set up a minimum cluster configuration. This configuration does not ensure data integrity under all failure conditions, because it does not include power switches. Note that this is a sample configuration; it is possible to set up a minimum configuration using other hardware.

Warning

The minimum cluster configuration is not a supported solution and *should not be used* in a production environment, as it does not ensure data integrity under all failure conditions.

Hardware	Description
At least two server systems	Each system becomes a node exclusively for use in the cluster; system hardware requirements are similar to that of Red Hat Enterprise Linux 4.
One network interface card (NIC) for each node	One network interface connects to a hub or switch for cluster connectivity.
Network cables with RJ45 connectors	Network cables connect to the network interface on each node for client access and heartbeat packets.
RAID storage enclosure	The RAID storage enclosure contains one controller with at least two host ports.
Two HD68 SCSI cables	Each cable connects one host bus adapter to one port on the RAID controller, creating two single-initiator SCSI buses.

Table 2-1. Example of Minimum Cluster Configuration

The minimum hardware configuration is a cost-effective cluster configuration for development purposes; however, it contains components that can cause service outages if failed. For example, if the RAID controller fails, then all cluster services become unavailable.

To improve availability, protect against component failure, and ensure data integrity under all failure conditions, more hardware is required. Refer to Table 2-2.

Problem	Solution
Disk failure	Hardware RAID to replicate data across multiple disks
RAID controller failure	Dual RAID controllers to provide redundant access to disk data
Network interface failure	Ethernet channel bonding and failover
Power source failure	Redundant uninterruptible power supply (UPS) systems
Machine failure	Power switches

Table 2-2. Improving Availability and Data Integrity

Figure 2-1 illustrates a hardware configuration with improved availability. This configuration uses a fence device (in this case, a network-attached power switch) and the nodes are configured for Red Hat GFS storage attached to a Fibre Channel SAN switch. For more information about configuring and using Red Hat GFS, refer to the *Red Hat GFS Administrator's Guide*.

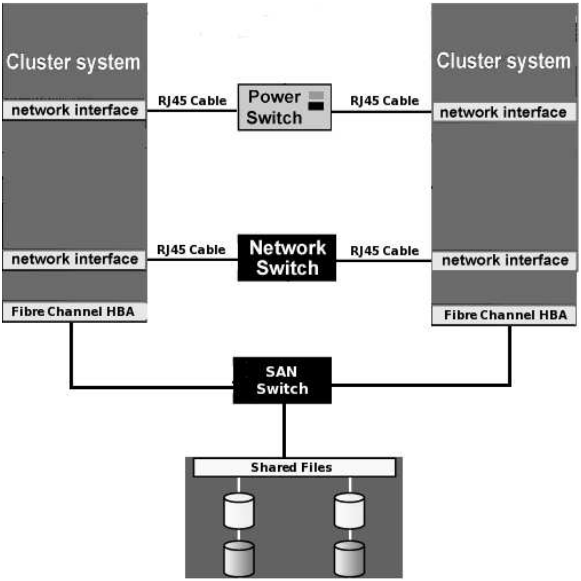


Figure 2-1. Hardware Configuration for Improved availability

A hardware configuration that ensures data integrity under failure conditions can include the following components:

- At least two servers to run cluster services
- Switched Ethernet connection between each node for heartbeat pings and for client network access
- Dual-controller RAID array or redundant access to SAN or other storage.

- Network power switches to enable each node to power-cycle the other nodes during the failover process
- Ethernet interfaces configured to use channel bonding
- At least two UPS systems for a highly-available source of power

The components described in Table 2-3 can be used to set up a no single point of failure cluster configuration that includes two single-initiator SCSI buses and power switches to ensure data integrity under all failure conditions. Note that this is a sample configuration; it is possible to set up a no single point of failure configuration using other hardware.

Hardware	Description
Two servers (up to 16 supported)	Each node includes the following hardware: Two network interfaces for: Client network access Fence device connection
One network switch	A network switch enables the connection of multiple nodes to a network.
Three network cables (each node)	Two cables to connect each node to the redundant network switches and a cable to connect to the fence device.
Two RJ45 to DB9 crossover cables	RJ45 to DB9 crossover cables connect a serial port on each node to the Cyclades terminal server.
Two power switches	Power switches enable each node to power-cycle the other node before restarting its services. Two RJ45 Ethernet cables for a node are connected to each switch.
FlashDisk RAID Disk Array with dual controllers	Dual RAID controllers protect against disk and controller failure. The RAID controllers provide simultaneous access to all the logical units on the host ports.
Two HD68 SCSI cables	HD68 cables connect each host bus adapter to a RAID enclosure "in" port, creating two single-initiator SCSI buses.
Two terminators	Terminators connected to each "out" port on the RAID enclosure terminate both single-initiator SCSI buses.
Redundant UPS Systems	UPS systems provide a highly-available source of power. The power cables for the power switches and the RAID enclosure are connected to two UPS systems.

Table 2-3. Example of a No Single Point of Failure Configuration

Cluster hardware configurations can also include other optional hardware components that are common in a computing environment. For example, a cluster can include a *network* *switch* or *network hub*, which enables the connection of the nodes to a network. A cluster may also include a *console switch*, which facilitates the management of multiple nodes and eliminates the need for separate monitors, mouses, and keyboards for each node.

One type of console switch is a *terminal server*, which enables connection to serial consoles and management of many nodes from one remote location. As a low-cost alternative, you can use a *KVM* (keyboard, video, and mouse) switch, which enables multiple nodes to share one keyboard, monitor, and mouse. A KVM switch is suitable for configurations in which access to a graphical user interface (GUI) to perform system management tasks is preferred.

When choosing a system, be sure that it provides the required PCI slots, network slots, and serial ports. For example, a no single point of failure configuration requires multiple bonded Ethernet ports. Refer to Section 2.3.1 *Installing the Basic Cluster Hardware* for more information.

2.1.2. Choosing the Type of Fence Device

The Red Hat Cluster Manager implementation consists of a generic power management layer and a set of device-specific modules which accommodate a range of power management types. When selecting the appropriate type of fence device to deploy in the cluster, it is important to recognize the implications of specific device types.

Important

> Use of a fencing method is an integral part of a production cluster environment. Configuration of a cluster without a fence device is not supported.

Red Hat Cluster Manager supports several types of fencing methods, including network power switches, fabric switches, and Integrated Power Management hardware. Table 2-5 summarizes the supported types of fence devices and some examples of brands and models that have been tested with Red Hat Cluster Manager.

Ultimately, choosing the right type of fence device to deploy in a cluster environment depends on the data integrity requirements versus the cost and availability of external power switches.

2.2. Cluster Hardware Components

Use the following section to identify the hardware components required for the cluster configuration.

Hardware	Quantity	Description	Required
Cluster nodes	16 (maximum supported)	Each node must provide enough PCI slots, network slots, and storage adapters for the cluster hardware configuration. Because attached storage devices must have the same device special file on each node, it is recommended that the nodes have symmetric I/O subsystems. It is also recommended that the processor speed and amount of system memory be adequate for the processes run on the cluster nodes. Refer to Section 2.3.1 *Installing the Basic Cluster Hardware* for more information.	Yes

Table 2-4. Cluster Node Hardware

Table 2-5 includes several different types of fence devices.

A single cluster requires only one type of power switch.

Type	Description	Models
Network-attached power switches.	Remote (LAN, Internet) fencing using RJ45 Ethernet connections and remote terminal access to the device.	APC MasterSwitch 92xx/96xx; WTI NPS-115/NPS-230, IPS-15, IPS-800/IPS-800-CE and TPS-2
Fabric Switches.	Fence control interface integrated in several models of fabric switches used for Storage Area Networks (SANs). Used as a way to fence a failed node from accessing shared data.	Brocade Silkworm 2x00, McData Sphereon, Vixel 9200
Integrated Power Management Interfaces	Remote power management features in various brands of server systems; can be used as a fencing agent in cluster systems	HP Integrated Lights-out (iLO), IBM BladeCenter with firmware dated 7-22-04 or later

Table 2-5. Fence Devices

Table 2-7 through Table 2-8 show a variety of hardware components for an administrator to choose from. An individual cluster does *not* require all of the components listed in these tables.

Hardware	Quantity	Description	Required
Network interface	One for each network connection	Each network connection requires a network interface installed in a node.	Yes
Network switch or hub	One	A network switch or hub allows connection of multiple nodes to a network.	Yes
Network cable	One for each network interface	A conventional network cable, such as a cable with an RJ45 connector, connects each network interface to a network switch or a network hub.	Yes

Table 2-6. Network Hardware Table

Hardware	Quantity	Description	Required
Host bus adapter	One per node	To connect to shared disk storage, install either a parallel SCSI or a Fibre Channel host bus adapter in a PCI slot in each cluster node. For parallel SCSI, use a low voltage differential (LVD) host bus adapter. Adapters have either HD68 or VHDCI connectors.	Yes

Hardware	Quantity	Description	Required
External disk storage enclosure	At least one	Use Fibre Channel or single-initiator parallel SCSI to connect the cluster nodes to a single or dual-controller RAID array. To use single-initiator buses, a RAID controller must have multiple host ports and provide simultaneous access to all the logical units on the host ports. To use a dual-controller RAID array, a logical unit must fail over from one controller to the other in a way that is transparent to the operating system. SCSI RAID arrays that provide simultaneous access to all logical units on the host ports are recommended. To ensure symmetry of device IDs and LUNs, many RAID arrays with dual redundant controllers must be configured in an active/passive mode. Refer to Appendix A *Supplementary Hardware Information* for more information.	Yes
SCSI cable	One per node	SCSI cables with 68 pins connect each host bus adapter to a storage enclosure port. Cables have either HD68 or VHDCI connectors. Cables vary based on adapter type.	Only for parallel SCSI configurations
SCSI terminator	As required by hardware configuration	For a RAID storage enclosure that uses "out" ports (such as FlashDisk RAID Disk Array) and is connected to single-initiator SCSI buses, connect terminators to the "out" ports to terminate the buses.	Only for parallel SCSI configurations and only as necessary for termination
Fibre Channel hub or switch	One or two	A Fibre Channel hub or switch may be required.	Only for some Fibre Channel configurations

Hardware	Quantity	Description	Required
Fibre Channel cable	As required by hardware configuration	A Fibre Channel cable connects a host bus adapter to a storage enclosure port, a Fibre Channel hub, or a Fibre Channel switch. If a hub or switch is used, additional cables are needed to connect the hub or switch to the storage adapter ports.	Only for Fibre Channel configurations

Table 2-7. Shared Disk Storage Hardware Table

Hardware	Quantity	Description	Required
UPS system	One or more	*Uninterruptible power supply* (UPS) systems protect against downtime if a power outage occurs. UPS systems are highly recommended for cluster operation. Connect the power cables for the shared storage enclosure and both power switches to redundant UPS systems. Note that a UPS system must be able to provide voltage for an adequate period of time, and should be connected to its own power circuit.	Strongly recommended for availability

Table 2-8. UPS System Hardware Table

Hardware	Quantity	Description	Required
Terminal server	One	A terminal server enables you to manage many nodes remotely.	No
KVM switch	One	A KVM switch enables multiple nodes to share one keyboard, monitor, and mouse. Cables for connecting nodes to the switch depend on the type of KVM switch.	No

Table 2-9. Console Switch Hardware Table

2.3. Setting Up the Nodes

After identifying the cluster hardware components described in Section 2.1 *Choosing a Hardware Configuration*, set up the basic cluster hardware and connect the nodes to the optional console switch and network switch or hub. Follow these steps:

1. In all nodes, install the required network adapters and host bus adapters. Refer to Section 2.3.1 *Installing the Basic Cluster Hardware* for more information about performing this task.
2. Set up the optional console switch and connect it to each node. Refer to Section 2.3.3 *Setting Up a Console Switch* for more information about performing this task.

   If a console switch is not used, then connect each node to a console terminal.
3. Set up the network switch or hub and use network cables to connect it to the nodes and the terminal server (if applicable). Refer to Section 2.3.4 *Setting Up a Network Switch or Hub* for more information about performing this task.

After performing the previous tasks, install Red Hat Enterprise Linux as described in Section 2.4 *Installing and Configuring Red Hat Enterprise Linux*.

2.3.1. Installing the Basic Cluster Hardware

Nodes must provide the CPU processing power and memory required by applications.

In addition, nodes must be able to accommodate the SCSI or Fibre Channel adapters, network interfaces, and serial ports that the hardware configuration requires. Systems have a limited number of pre-installed serial and network ports and PCI expansion slots. Table 2-10 helps determine how much capacity the employed node systems require.

Cluster Hardware Component	Serial Ports	Ethernet Ports	PCI Slots
SCSI or Fibre Channel adapter to shared disk storage			One for each bus adapter
Network connection for client access and Ethernet heartbeat pings		One for each network connection	

Cluster Hardware Component	Serial Ports	Ethernet Ports	PCI Slots
Point-to-point Ethernet connection for 2-node clusters (optional)		One for each connection	
Terminal server connection (optional)	One		

Table 2-10. Installing the Basic Cluster Hardware

Most systems come with at least one serial port. If a system has graphics display capability, it is possible to use the serial console port for a power switch connection. To expand your serial port capacity, use multi-port serial PCI cards. For multiple-node clusters, use a network power switch.

Also, ensure that local system disks are not on the same SCSI bus as the shared disks. For example, use two-channel SCSI adapters, such as the Adaptec 39160-series cards, and put the internal devices on one channel and the shared disks on the other channel. Using multiple SCSI cards is also possible.

Refer to the system documentation supplied by the vendor for detailed installation information. Refer to Appendix A *Supplementary Hardware Information* for hardware-specific information about using host bus adapters in a cluster.

2.3.2. Shared Storage considerations

In a cluster, shared disks can be used to store cluster service data. Because this storage must be available to all nodes running the cluster service configured to use the storage, it cannot be located on disks that depend on the availability of any one node.

There are some factors to consider when setting up shared disk storage in a cluster:

- It is recommended to use a clustered file system such as Red Hat GFS to configure Red Hat Cluster Manager storage resources, as it offers shared storage that is suited for high-availability cluster services. For more information about installing and configuring Red Hat GFS, refer to the *Red Hat GFS Administrator's Guide*.
- Whether you are using Red Hat GFS, local, or remote (for example, NFS) storage, it is *strongly recommended* that you connect any storage systems or enclosures to redundant UPS systems for a highly-available source of power. Refer to Section 2.5.3 *Configuring UPS Systems* for more information.
- The use of software RAID or *Logical Volume Management* (LVM) for shared storage is not supported. This is because these products do not coordinate access to shared storage from multiple hosts. Software RAID or LVM may be used on non-shared storage on

cluster nodes (for example, boot and system partitions, and other file systems that are not associated with any cluster services).

An exception to this rule is *CLVM*, the daemon and library that supports clustering of LVM2. CLVM allows administrators to configure shared storage for use as a resource in cluster services when used in conjunction with the CMAN cluster manager and the *Distributed Lock Manager* (DLM) mechanism for prevention of simultaneous node access to data and possible corruption. In addition, CLVM works with GULM as its cluster manager and lock manager.

- For remote file systems such as NFS, you may use gigabit Ethernet for improved bandwidth over 10/100 Ethernet connections. Consider redundant links or channel bonding for improved remote file system availability. Refer to Section 2.5.1 *Configuring Ethernet Channel Bonding* for more information.
- Multi-initiator SCSI configurations are not supported due to the difficulty in obtaining proper bus termination. Refer to Appendix A *Supplementary Hardware Information* for more information about configuring attached storage.
- A shared partition can be used by only one cluster service.
- Do not include any file systems used as a resource for a cluster service in the node's local `/etc/fstab` files, because the cluster software must control the mounting and unmounting of service file systems.
- For optimal performance of shared file systems, make sure to specify a 4 KB block size with the `mke2fs -b` command. A smaller block size can cause long `fsck` times. Refer to Section 2.5.3.2 *Creating File Systems*.

After setting up the shared disk storage hardware, partition the disks and create file systems on the partitions. Refer to Section 2.5.3.1 *Partitioning Disks*, and Section 2.5.3.2 *Creating File Systems* for more information on configuring disks.

2.3.3. Setting Up a Console Switch

Although a console switch is not required for cluster operation, it can be used to facilitate node management and eliminate the need for separate monitors, mouses, and keyboards for each cluster node. There are several types of console switches.

For example, a terminal server enables connection to serial consoles and management of many nodes from a remote location. For a low-cost alternative, use a KVM (keyboard, video, and mouse) switch, which enables multiple nodes to share one keyboard, monitor, and mouse. A KVM switch is suitable for configurations in which GUI access to perform system management tasks is preferred.

Set up the console switch according to the documentation provided by the vendor.

After the console switch has been set up, connect it to each cluster node. The cables used depend on the type of console switch. For example, a Cyclades terminal server uses RJ45 to DB9 crossover cables to connect a serial port on each node to the terminal server.

2.3.4. Setting Up a Network Switch or Hub

A network switch or hub, although not required for operating a two-node cluster, can be used to facilitate cluster and client system network operations. Clusters of more than two nodes require a switch or hub.

Set up a network switch or hub according to the documentation provided by the vendor.

After setting up the network switch or hub, connect it to each node by using conventional network cables. A terminal server, if used, is connected to the network switch or hub through a network cable.

2.4. Installing and Configuring Red Hat Enterprise Linux

After the setup of basic cluster hardware, proceed with installation of Red Hat Enterprise Linux on each node and ensure that all systems recognize the connected devices. Follow these steps:

1. Install Red Hat Enterprise Linux on all cluster nodes. Refer to *Red Hat Enterprise Linux Installation Guide* for instructions.

   In addition, when installing Red Hat Enterprise Linux, it is *strongly recommended* to do the following:

   - Gather the IP addresses for the nodes and for the bonded Ethernet ports, before installing Red Hat Enterprise Linux. Note that the IP addresses for the bonded Ethernet ports can be private IP addresses, (for example, 10.*x.x.x*).
   - Do not place local file systems (such as /, /etc, /tmp, and /var) on shared disks or on the same SCSI bus as shared disks. This helps prevent the other cluster nodes from accidentally mounting these file systems, and also reserves the limited number of SCSI identification numbers on a bus for cluster disks.
   - Place /tmp and /var on different file systems. This may improve node performance.
   - When a node boots, be sure that the node detects the disk devices in the same order in which they were detected during the Red Hat Enterprise Linux installation. If the devices are not detected in the same order, the node may not boot.
   - When using certain RAID storage configured with Logical Unit Numbers (LUNs) greater than zero, it may be necessary to enable LUN support by adding the following to /etc/modprobe.conf:

     ```
     options scsi_mod max_scsi_luns=255
     ```

2. Reboot the nodes.

3. When using a terminal server, configure Red Hat Enterprise Linux to send console messages to the console port.
4. Edit the `/etc/hosts` file on each cluster node and include the IP addresses used in the cluster or ensure that the addresses are in DNS. Refer to Section 2.4.1 *Editing the `/etc/hosts` File* for more information about performing this task.
5. Decrease the alternate kernel boot timeout limit to reduce boot time for nodes. Refer to Section 2.4.2 *Decreasing the Kernel Boot Timeout Limit* for more information about performing this task.
6. Ensure that no login (or `getty`) programs are associated with the serial ports that are being used for the remote power switch connection (if applicable). To perform this task, edit the `/etc/inittab` file and use a hash symbol (#) to comment out the entries that correspond to the serial ports used for the remote power switch. Then, invoke the `init q` command.
7. Verify that all systems detect all the installed hardware:
    - Use the `dmesg` command to display the console startup messages. Refer to Section 2.4.3 *Displaying Console Startup Messages* for more information about performing this task.
    - Use the `cat /proc/devices` command to display the devices configured in the kernel. Refer to Section 2.4.4 *Displaying Devices Configured in the Kernel* for more information about performing this task.
8. Verify that the nodes can communicate over all the network interfaces by using the `ping` command to send test packets from one node to another.
9. If intending to configure Samba services, verify that the required RPM packages for Samba services are installed.

2.4.1. Editing the `/etc/hosts` File

The `/etc/hosts` file contains the IP address-to-hostname translation table. The `/etc/hosts` file on each node must contain entries for IP addresses and associated hostnames for all cluster nodes.

As an alternative to the `/etc/hosts` file, name services such as DNS or NIS can be used to define the host names used by a cluster. However, to limit the number of dependencies and optimize availability, it is strongly recommended to use the `/etc/hosts` file to define IP addresses for cluster network interfaces.

The following is an example of an /etc/hosts file on a node of a cluster that does not use DNS-assigned hostnames:

```
127.0.0.1         localhost.localdomain      localhost
192.168.1.81      node1.example.com          node1
193.186.1.82      node2.example.com          node2
193.186.1.83      node3.example.com          node3
```

The previous example shows the IP addresses and hostnames for three nodes (*node1*, *node2*, and *node3*),

Important

Do *not* assign the node hostname to the localhost (127.0.0.1) address, as this causes issues with the CMAN cluster management system.

Verify correct formatting of the local host entry in the /etc/hosts file to ensure that it does not include non-local systems in the entry for the local host. An example of an incorrect local host entry that includes a non-local system (*server1*) is shown next:

```
127.0.0.1     localhost.localdomain     localhost server1
```

An Ethernet connection may not operate properly if the format of the /etc/hosts file is not correct. Check the /etc/hosts file and correct the file format by removing non-local systems from the local host entry, if necessary.

Note that each network adapter must be configured with the appropriate IP address and netmask.

The following example shows a portion of the output from the /sbin/ip addr list command on a cluster node:

```
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
       valid_lft forever preferred_lft forever
```

You may also add the IP addresses for the cluster nodes to your DNS server. Refer to the *Red Hat Enterprise Linux System Administration Guide* for information on configuring DNS, or consult your network administrator.

2.4.2. Decreasing the Kernel Boot Timeout Limit

It is possible to reduce the boot time for a node by decreasing the kernel boot timeout limit. During the Red Hat Enterprise Linux boot sequence, the boot loader allows for specifying an alternate kernel to boot. The default timeout limit for specifying a kernel is ten seconds.

To modify the kernel boot timeout limit for a node, edit the appropriate files as follows:

When using the GRUB boot loader, the timeout parameter in `/boot/grub/grub.conf` should be modified to specify the appropriate number of seconds for the *`timeout`* parameter. To set this interval to 3 seconds, edit the parameter to the following:

```
timeout = 3
```

When using the LILO or ELILO boot loaders, edit the `/etc/lilo.conf` file (on x86 systems) or the `elilo.conf` file (on Itanium systems) and specify the desired value (in tenths of a second) for the *`timeout`* parameter. The following example sets the timeout limit to three seconds:

```
timeout = 30
```

To apply any changes made to the `/etc/lilo.conf` file, invoke the `/sbin/lilo` command.

On an Itanium system, to apply any changes made to the `/boot/efi/efi/redhat/elilo.conf` file, invoke the `/sbin/elilo` command.

2.4.3. Displaying Console Startup Messages

Use the `dmesg` command to display the console startup messages. Refer to the dmesg(8) man page for more information.

The following example of output from the `dmesg` command shows that two external SCSI buses and nine disks were detected on the node. (Lines with backslashes display as one line on most screens):

```
May 22 14:02:10 storage3 kernel: scsi0 : Adaptec AHA274x/284x/294x \
      (EISA/VLB/PCI-Fast SCSI) 5.1.28/3.2.4
May 22 14:02:10 storage3 kernel:
May 22 14:02:10 storage3 kernel: scsi1 : Adaptec AHA274x/284x/294x \
            (EISA/VLB/PCI-Fast SCSI) 5.1.28/3.2.4
May 22 14:02:10 storage3 kernel:
May 22 14:02:10 storage3 kernel: scsi : 2 hosts.
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST39236LW        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sda at scsi0, channel 0, id 0,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC       Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdb at scsi1, channel 0, id 0,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC       Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdc at scsi1, channel 0, id 1,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdd at scsi1, channel 0, id 2,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sde at scsi1, channel 0, id 3,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdf at scsi1, channel 0, id 8,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdg at scsi1, channel 0, id 9,
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdh at scsi1, channel 0, id 10
May 22 14:02:11 storage3 kernel:   Vendor: SEAGATE   Model: ST318203LC        Rev:
May 22 14:02:11 storage3 kernel: Detected scsi disk sdi at scsi1, channel 0, id 11
May 22 14:02:11 storage3 kernel:   Vendor: Dell      Model: 8 BAY U2W CU      Rev:
May 22 14:02:11 storage3 kernel:   Type:   Processor \
                          ANSI SCSI revision: 03
May 22 14:02:11 storage3 kernel: scsi1 : channel 0 target 15 lun 1 request sense \
       failed, performing reset.
May 22 14:02:11 storage3 kernel: SCSI bus is being reset for host 1 channel 0.
May 22 14:02:11 storage3 kernel: scsi : detected 9 SCSI disks total.
```

The following example of the `dmesg` command output shows that a quad Ethernet card was detected on the node:

```
May 22 14:02:11 storage3 kernel: 3c59x.c:v0.99H 11/17/98 Donald Becker
May 22 14:02:11 storage3 kernel: tulip.c:v0.91g-ppc 7/16/99
May 22 14:02:11 storage3 kernel: eth0: Digital DS21140 Tulip rev 34 at 0x9800, \
       00:00:BC:11:76:93, IRQ 5.
May 22 14:02:12 storage3 kernel: eth1: Digital DS21140 Tulip rev 34 at 0x9400, \
       00:00:BC:11:76:92, IRQ 9.
May 22 14:02:12 storage3 kernel: eth2: Digital DS21140 Tulip rev 34 at 0x9000, \
       00:00:BC:11:76:91, IRQ 11.
May 22 14:02:12 storage3 kernel: eth3: Digital DS21140 Tulip rev 34 at 0x8800, \
       00:00:BC:11:76:90, IRQ 10.
```

2.4.4. Displaying Devices Configured in the Kernel

To be sure that the installed devices (such as network interfaces), are configured in the kernel, use the `cat /proc/devices` command on each node. For example:

```
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  6 lp
  7 vcs
 10 misc
 13 input
 14 sound
 29 fb
 89 i2c
116 alsa
128 ptm
136 pts
171 ieee1394
180 usb
216 rfcomm
226 drm
254 pcmcia

Block devices:
  1 ramdisk
  2 fd
  3 ide0
  8 sd
  9 md
 65 sd
 66 sd
 67 sd
 68 sd
 69 sd
 70 sd
 71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
253 device-mapper
```

The previous example shows:

- Onboard serial ports (`ttyS`)
- USB devices (`usb`)
- SCSI devices (`sd`)

2.5. Setting Up and Connecting the Cluster Hardware

After installing Red Hat Enterprise Linux, set up the cluster hardware components and verify the installation to ensure that the nodes recognize all the connected devices. Note that the exact steps for setting up the hardware depend on the type of configuration. Refer to Section 2.1 *Choosing a Hardware Configuration* for more information about cluster configurations.

To set up the cluster hardware, follow these steps:

1. Shut down the nodes and disconnect them from their power source.
2. When using power switches, set up the switches and connect each node to a power switch. Refer to Section 2.5.2 *Configuring a Fence Device* for more information.

   In addition, it is recommended to connect each power switch (or each node's power cord if not using power switches) to a different UPS system. Refer to Section 2.5.3 *Configuring UPS Systems* for information about using optional UPS systems.
3. Set up shared disk storage according to the vendor instructions and connect the nodes to the external storage enclosure. Refer to Section 2.3.2 *Shared Storage considerations*.

   In addition, it is recommended to connect the storage enclosure to redundant UPS systems. Refer to Section 2.5.3 *Configuring UPS Systems* for more information about using optional UPS systems.
4. Turn on power to the hardware, and boot each cluster node. During the boot-up process, enter the BIOS utility to modify the node setup, as follows:

   - Ensure that the SCSI identification number used by the host bus adapter is unique for the SCSI bus it is attached to. Refer to Section A.3.4 *SCSI Identification Numbers* for more information about performing this task.
   - Enable or disable the onboard termination for each host bus adapter, as required by the storage configuration. Refer to Section A.3.2 *SCSI Bus Termination* for more information about performing this task.
   - Enable the node to automatically boot when it is powered on.

5. Exit from the BIOS utility, and continue to boot each node. Examine the startup messages to verify that the Red Hat Enterprise Linux kernel has been configured and can recognize the full set of shared disks. Use the `dmesg` command to display console startup messages. Refer to Section 2.4.3 *Displaying Console Startup Messages* for more information about using the `dmesg` command.

6. Set up the bonded Ethernet channels, if applicable. Refer to Section 2.5.1 *Configuring Ethernet Channel Bonding* for more information.
7. Run the `ping` command to verify packet transmission between *all* cluster nodes.

2.5.1. Configuring Ethernet Channel Bonding

Ethernet channel bonding in a no-single-point-of-failure cluster system allows for a fault tolerant network connection by combining two Ethernet devices into one virtual device. The resulting channel bonded interface ensures that in the event that one Ethernet device fails, the other device will become active. This type of channel bonding, called an *active-backup* policy allows connection of both bonded devices to one switch or can allow each Ethernet device to be connected to separate hubs or switches, which eliminates the single point of failure in the network hub/switch.

Channel bonding requires each cluster node to have two Ethernet devices installed. When it is loaded, the bonding module uses the MAC address of the first enslaved network device and assigns that MAC address to the other network device if the first device fails link detection.

To configure two network devices for channel bonding, perform the following:

1. Create a bonding devices in `/etc/modprobe.conf`. For example:

   ```
   alias bond0 bonding
   options bonding miimon=100 mode=1
   ```

   This loads the bonding device with the `bond0` interface name, as well as passes options to the bonding driver to configure it as an active-backup master device for the enslaved network interfaces.

2. Edit the `/etc/sysconfig/network-scripts/ifcfg-ethX` configuration file for both eth0 and eth1 so that the files show identical contents. For example:

   ```
   DEVICE=ethX
   USERCTL=no
   ONBOOT=yes
   MASTER=bond0
   SLAVE=yes
   BOOTPROTO=none
   ```

   This will enslave eth*X* (replace *X* with the assigned number of the Ethernet devices) to the bond0 master device.

3. Create a network script for the bonding device (for example, `/etc/sysconfig/network-scripts/ifcfg-bond0`), which would appear like the following example:

   ```
   DEVICE=bond0
   USERCTL=no
   ONBOOT=yes
   BROADCAST=192.168.1.255
   NETWORK=192.168.1.0
   ```

```
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
IPADDR=192.168.1.10
```

4. Reboot the system for the changes to take effect.

2.5.2. Configuring a Fence Device

Fence devices enable a node to power-cycle another node before restarting its services as part of the failover process. The ability to remotely disable a node ensures data integrity is maintained under any failure condition. Deploying a cluster in a production environment *requires* the use of a fence device. Only development (test) environments should use a configuration without a fence device. Refer to Section 2.1.2 *Choosing the Type of Fence Device* for a description of the various types of power switches.

In a cluster configuration that uses fence devices such as power switches, each node is connected to a switch through either a serial port (for two-node clusters) or network connection (for multi-node clusters). When failover occurs, a node can use this connection to power-cycle another node before restarting its services.

Fence devices protect against data corruption if an unresponsive (or hanging) node becomes responsive after its services have failed over, and issues I/O to a disk that is also receiving I/O from another node. In addition, if CMAN detects node failure, the failed node will be removed from the cluster. If a fence device is not used in the cluster, then a failed node may result in cluster services being run on more than one node, which can cause data corruption and possibly system crashes.

A node may appear to *hang* for a few seconds if it is swapping or has a high system workload. For this reason, adequate time is allowed prior to concluding that a node has failed.

If a node fails, and a fence device is used in the cluster, the fencing daemon power-cycles the hung node before restarting its services. This causes the hung node to reboot in a clean state and prevent it from issuing I/O and corrupting cluster service data.

When used, fence devices must be set up according to the vendor instructions; however, some cluster-specific tasks may be required to use them in a cluster. Consult the manufacturer documentation on configuring the fence device. Note that the cluster-specific information provided in this manual supersedes the vendor information.

When cabling a physical fence device such as a power switch, take special care to ensure that each cable is plugged into the appropriate port and configured correctly. This is crucial because there is no independent means for the software to verify correct cabling. Failure to cable correctly can lead to an incorrect node being power cycled, fenced off from shared storage via fabric-level fencing, or for a node to inappropriately conclude that it has successfully power cycled a failed node.

2.5.3. Configuring UPS Systems

Uninterruptible power supplies (UPS) provide a highly-available source of power. Ideally, a redundant solution should be used that incorporates multiple UPS systems (one per server). For maximal fault-tolerance, it is possible to incorporate two UPS systems per server as well as APC Automatic Transfer Switches to manage the power and shutdown management of the server. Both solutions are solely dependent on the level of availability desired.

It is not recommended to use a single UPS infrastructure as the sole source of power for the cluster. A UPS solution dedicated to the cluster is more flexible in terms of manageability and availability.

A complete UPS system must be able to provide adequate voltage and current for a prolonged period of time. While there is no single UPS to fit every power requirement, a solution can be tailored to fit a particular configuration.

If the cluster disk storage subsystem has two power supplies with separate power cords, set up two UPS systems, and connect one power switch (or one node's power cord if not using power switches) and one of the storage subsystem's power cords to each UPS system. A redundant UPS system configuration is shown in Figure 2-2.

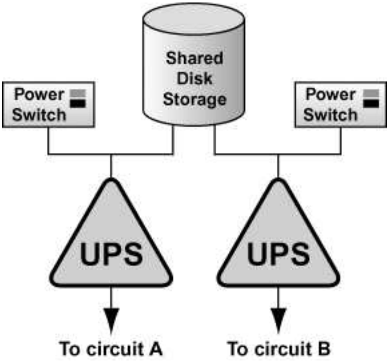


Figure 2-2. Redundant UPS System Configuration

An alternative redundant power configuration is to connect the power switches (or the nodes' power cords) and the disk storage subsystem to the same UPS system. This is the most cost-effective configuration, and provides some protection against power failure. However, if a power outage occurs, the single UPS system becomes a possible single point of failure. In addition, one UPS system may not be able to provide enough power to all the attached devices for an adequate amount of time. A single UPS system configuration is shown in Figure 2-3.

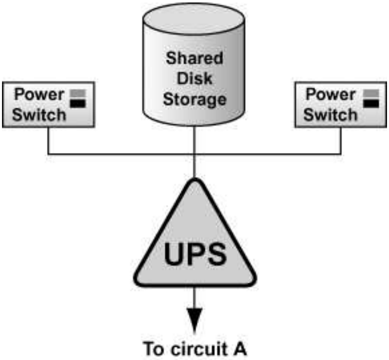


Figure 2-3. Single UPS System Configuration

Many vendor-supplied UPS systems include Red Hat Enterprise Linux applications that monitor the operational status of the UPS system through a serial port connection. If the battery power is low, the monitoring software initiates a clean system shutdown. As this occurs, the cluster software is properly stopped, because it is controlled by a SysV runlevel script (for example, `/etc/rc.d/init.d/rgmanager`).

Refer to the UPS documentation supplied by the vendor for detailed installation information.

2.5.3.1. Partitioning Disks

After shared disk storage has been set up, partition the disks so they can be used in the cluster. Then, create file systems or raw devices on the partitions.

Use `parted` to modify a disk partition table and divide the disk into partitions. While in `parted`, use the `p` to display the partition table and the `mkpart` command to create new partitions. The following example shows how to use `parted` to create a partition on disk:

- Invoke `parted` from the shell using the command `parted` and specifying an available shared disk device. At the `(parted)` prompt, use the `p` to display the current partition table. The output should be similar to the following:

  ```
  Disk geometry for /dev/sda: 0.000-4340.294 megabytes
  Disk label type: msdos
  Minor    Start       End     Type      Filesystem  Flags
  ```

- Decide on how large of a partition is required. Create a partition of this size using the `mkpart` command in `parted`. Although the `mkpart` does not create a file system, it normally requires a file system type at partition creation time. `parted` uses a range on the disk to determine partition size; the size is the space between the end and the beginning of the given range. The following example shows how to create two partitions of 20 MB each on an empty disk.

```
(parted) mkpart primary ext3 0 20
(parted) mkpart primary ext3 20 40
(parted) p
Disk geometry for /dev/sda: 0.000-4340.294 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.030     21.342  primary
2         21.343     38.417  primary
```

- When more than four partitions are required on a single disk, it is necessary to create an *extended partition*. If an extended partition is required, the `mkpart` also performs this task. In this case, it is not necessary to specify a file system type.

Note

Only one extended partition may be created, and the extended partition *must* be one of the four primary partitions.

```
(parted) mkpart extended 40 2000
(parted) p
Disk geometry for /dev/sda: 0.000-4340.294 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.030     21.342  primary
2         21.343     38.417  primary
3         38.417   2001.952  extended
```

- An extended partition allows the creation of *logical partitions*inside of it. The following example shows the division of the extended partition into two logical partitions.

```
(parted) mkpart logical ext3 40 1000
(parted) p
Disk geometry for /dev/sda: 0.000-4340.294 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.030     21.342  primary
2         21.343     38.417  primary
3         38.417   2001.952  extended
5         38.447    998.841  logical
(parted) mkpart logical ext3 1000 2000
(parted) p
Disk geometry for /dev/sda: 0.000-4340.294 megabytes
Disk label type: msdos
Minor    Start       End     Type      Filesystem  Flags
1          0.030     21.342  primary
2         21.343     38.417  primary
3         38.417   2001.952  extended
5         38.447    998.841  logical
```

```
6        998.872   2001.952  logical
```

- A partition may be removed using `parted`'s `rm` command. For example:

  ```
  (parted) rm 1
  (parted) p
  Disk geometry for /dev/sda: 0.000-4340.294 megabytes
  Disk label type: msdos
  Minor    Start        End     Type      Filesystem  Flags
  2         21.343     38.417  primary
  3         38.417   2001.952  extended
  5         38.447    998.841  logical
  6        998.872   2001.952  logical
  ```

- After all required partitions have been created, exit `parted` using the `quit` command. If a partition was added, removed, or changed while both nodes are powered on and connected to the shared storage, reboot the other node for it to recognize the modifications. After partitioning a disk, format the partition for use in the cluster. For example, create the file systems for shared partitions. Refer to Section 2.5.3.2 *Creating File Systems* for more information on configuring file systems.

  For basic information on partitioning hard disks at installation time, refer to the *Red Hat Enterprise Linux Installation Guide*.

2.5.3.2. Creating File Systems

Use the `mkfs` command to create an ext3 file system. For example:

```
mke2fs -j -b 4096 /dev/sde3
```

For optimal performance of shared file systems, make sure to specify a 4 KB block size with the `mke2fs -b` command. A smaller block size can cause long `fsck` times.

Chapter 3.

Installing and Configuring Red Hat Cluster Suite Software

This chapter describes how to install and configure Red Hat Cluster Suite software and consists of the following sections:

- Section 3.1 *Software Installation and Configuration Tasks*
- Section 3.2 *Overview of the* ***Cluster Configuration Tool***
- Section 3.3 *Installing the Red Hat Cluster Suite Packages*
- Section 3.4 *Starting the* ***Cluster Configuration Tool***
- Section 3.5 *Naming The Cluster*
- Section 3.6 *Configuring Fence Devices*
- Section 3.7 *Adding and Deleting Members*
- Section 3.8 *Configuring a Failover Domain*
- Section 3.9 *Adding Cluster Resources*
- Section 3.10 *Adding a Cluster Service to the Cluster*
- Section 3.11 *Propagating The Configuration File: New Cluster*
- Section 3.12 *Starting the Cluster Software*

3.1. Software Installation and Configuration Tasks

Installing and configuring Red Hat Cluster Suite software consists of the following steps:

1. Installing Red Hat Cluster Suite software.

   Refer to Section 3.3 *Installing the Red Hat Cluster Suite Packages*.

2. Starting the **Cluster Configuration Tool.**

   a. Creating a new configuration file or using an existing one.

   b. Choose locking: either DLM or GULM.

   Refer to Section 3.4 *Starting the* ***Cluster Configuration Tool***.

3. Naming the cluster. Refer to Section 3.5 *Naming The Cluster*.

4. Creating fence devices. Refer to Section 3.6 *Configuring Fence Devices*.

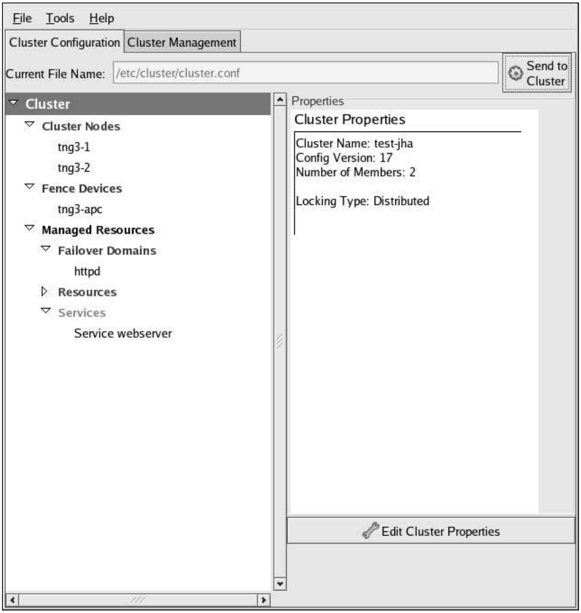
5. Creating cluster members. Refer to Section 3.7 *Adding and Deleting Members*.
6. Creating failover domains. Refer to Section 3.8 *Configuring a Failover Domain*.
7. Creating resources. Refer to Section 3.9 *Adding Cluster Resources*.
8. Creating cluster services.

   Refer to Section 3.10 *Adding a Cluster Service to the Cluster*.
9. Propagating the configuration file to the other nodes in the cluster.

   Refer to Section 3.11 *Propagating The Configuration File: New Cluster*.
10. Starting the cluster software. Refer to Section 3.12 *Starting the Cluster Software*.

3.2. Overview of the Cluster Configuration Tool

The **Cluster Configuration Tool** (Figure 3-1) is a graphical user interface (GUI) for creating, editing, saving, and propagating the cluster configuration file, `/etc/cluster/cluster.conf`. The **Cluster Configuration Tool** is part of the Red Hat Cluster Suite management GUI, (the `system-config-cluster` package) and is accessed by the **Cluster Configuration** tab in the Red Hat Cluster Suite management GUI.



Current File Name: /etc/cluster/cluster.conf

Cluster Properties

Cluster Name: test-jha
Config Version: 17
Number of Members: 2

Locking Type: Distributed

Figure 3-1. Cluster Configuration Tool

The **Cluster Configuration Tool** uses a hierarchical structure to show relationships among components in the cluster configuration. A triangle icon to the left of a component name indicates that the component has one or more subordinate components assigned to it. To expand or collapse the portion of the tree below a component, click the triangle icon.

The **Cluster Configuration Tool** represents the cluster configuration with the following components in the left frame:

- **Cluster Nodes** — Defines cluster nodes. Nodes are represented by name as subordinate elements under **Cluster Nodes**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can add nodes, delete nodes, edit node properties, and configure fencing methods for each node.
- **Fence Devices** — Defines fence devices. Fence devices are represented as subordinate elements under **Fence Devices**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can add fence devices, delete fence devices, and edit fence-device properties. Fence devices must be defined before you can configure fencing (with the **Manage Fencing For This Node** button) for each node.

- **Managed Resources** — Defines failover domains, resources, and services.
  - **Failover Domains** — Use this section to configure one or more subsets of cluster nodes used to run a service in the event of a node failure. Failover domains are represented as subordinate elements under **Failover Domains**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can create failover domains (when **Failover Domains** is selected) or edit failover domain properties (when a failover domain is selected).
  - **Resources** — Use this section to configure resources to be managed by the system. Choose from the available list of file systems, IP addresses, NFS mounts and exports, and user-created scripts and configure them individually. Resources are represented as subordinate elements under **Resources**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can create resources (when **Resources** is selected) or edit resource properties (when a resource is selected).
  - **Services** — Use this section to create and configure services that combine cluster resources, nodes, and failover domains as needed. Services are represented as subordinate elements under **Services**. Using configuration buttons at the bottom of the right frame (below **Properties**), you can create services (when **Services** is selected) or edit service properties (when a service is selected).

Warning

Do not manually edit the contents of the `/etc/cluster/cluster.conf` file without guidance from an authorized Red Hat representative or unless you fully understand the consequences of editing the `/etc/cluster/cluster.conf` file manually.

Figure 3-2 shows the hierarchical relationship among cluster configuration components. The cluster comprises cluster nodes. The cluster nodes are connected to one or more fencing devices. Nodes can be separated by failover domains to a cluster service. The services comprise managed resources such as NFS exports, IP addresses, and shared GFS partitions. The structure is ultimately reflected in the `/etc/cluster/cluster.conf` XML structure. The **Cluster Configuration Tool** provides a convenient way to create and manipulate the `/etc/cluster/cluster.conf` file.

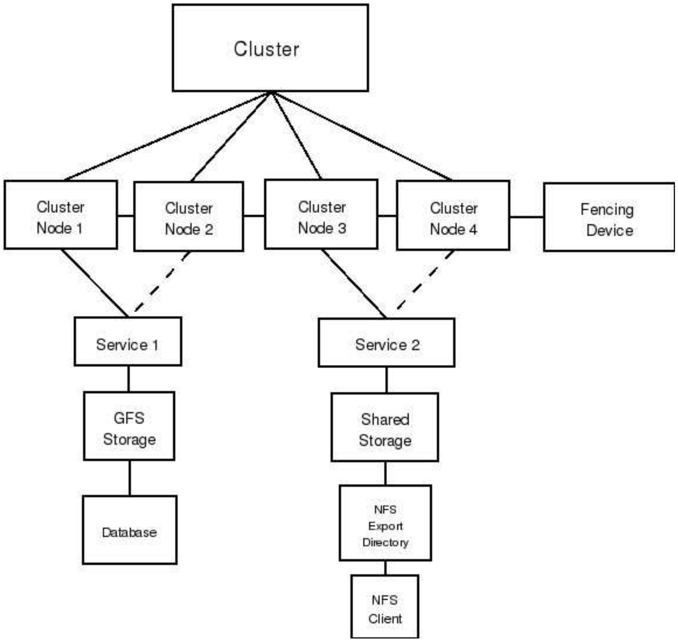


Figure 3-2. Cluster Configuration Structure

3.3. Installing the Red Hat Cluster Suite Packages

You can install Red Hat Cluster Suite and (optionally install) Red Hat GFS RPMs automatically by running the `up2date` utility at each node for the Red Hat Cluster Suite and Red Hat GFS products.

Tip

You can access the Red Hat Cluster Suite and Red Hat GFS products by using Red Hat Network to subscribe to and access the channels containing the Red Hat Cluster Suite and Red Hat GFS packages. From the Red Hat Network channel, you can manage entitlements for your cluster nodes and upgrade packages for each node within the Red Hat Network Web-based interface. For more information on using Red Hat Network, visit http://rhn.redhat.com.

To automatically install RPMs, follow these steps at each node:

1. Log on as the root user.
2. Run `up2date --installall --channel Label` for Red Hat Cluster Suite. The following example shows running the command for i386 RPMs:
   ```
   # up2date --installall --channel rhel-i386-as-4-cluster
   ```
3. (Optional) If you are installing Red Hat GFS, run `up2date --installall --channel Label` for Red Hat GFS. The following example shows running the command for i386 RPMs:
   ```
   # up2date --installall --channel rhel-i386-as-4-gfs-6.1
   ```

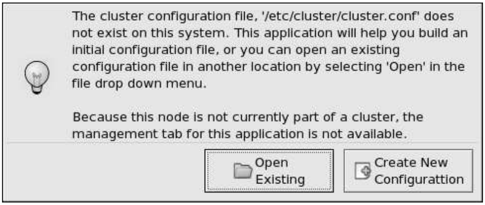
Note

The preceding procedure accommodates most installation requirements. However, if your installation has extreme limitations on storage and RAM, refer to Appendix B *Selectively Installing Red Hat Cluster Suite Packages* for more detailed information about Red Hat Cluster Suite and Red Hat GFS RPM packages and customized installation of those packages.

3.4. Starting the Cluster Configuration Tool

You can start the **Cluster Configuration Tool** by logging in to a cluster node as root with the `ssh -Y` command and issuing the `system-config-cluster` command. For example, to start the **Cluster Configuration Tool** on cluster node nano-01, do the following:

1. Log in to a cluster node and run `system-config-cluster`. For example:
   ```
   $ssh -Y root@nano-01
             .
             .
             .
   #system-config-cluster
   ```
   a. If this is the first time you have started the **Cluster Configuration Tool**, the program prompts you to either open an existing configuration or create a new one. Click **Create New Configuration** to start a new configuration file (refer to Figure 3-3).



The cluster configuration file, '/etc/cluster/cluster.conf' does not exist on this system. This application will help you build an initial configuration file, or you can open an existing configuration file in another location by selecting 'Open' in the file drop down menu.

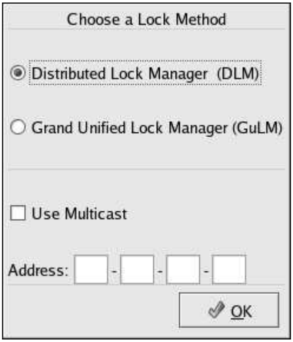
Because this node is not currently part of a cluster, the management tab for this application is not available.

Figure 3-3. Starting a New Configuration File

Note

The **Cluster Management** tab for the Red Hat Cluster Suite management GUI is available after you save the configuration file with the **Cluster Configuration Tool**, exit, and restart the the Red Hat Cluster Suite management GUI (`system-config-cluster`). (The **Cluster Management** tab displays the status of the cluster service manager, cluster nodes, and resources, and shows statistics concerning cluster service operation. To manage the cluster system further, choose the **Cluster Configuration** tab.)

b. For a new configuration, a **Lock Method** dialog box is displayed requesting a choice of either the GULM or DLM lock method (and multicast address for DLM).



Choose a Lock Method

● Distributed Lock Manager (DLM)

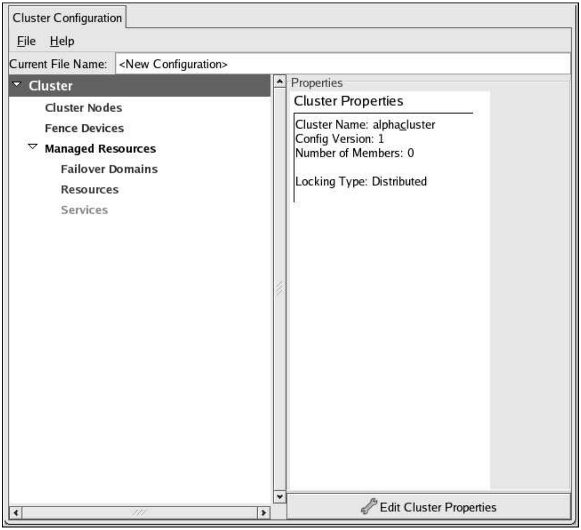
○ Grand Unified Lock Manager (GuLM)

□ Use Multicast

Address: _ - _ - _ - _

Figure 3-4. Choosing a Lock Method

2. Starting the **Cluster Configuration Tool** displays a graphical representation of the configuration (Figure 3-5) as specified in the cluster configuration file, `/etc/cluster/cluster.conf`.



Current File Name: <New Configuration>

Cluster Name: alphacluster
Config Version: 1
Number of Members: 0

Locking Type: Distributed

Figure 3-5. The Cluster Configuration Tool

3.5. Naming The Cluster

Naming the cluster consists of specifying a cluster name, a configuration version (optional), and values for **Post-Join Delay** and **Post-Fail Delay**. Name the cluster as follows:

1. At the left frame, click **Cluster**.
2. At the bottom of the right frame (labeled **Properties**), click the **Edit Cluster Properties** button. Clicking that button causes a **Cluster Properties** dialog box to be displayed. The **Cluster Properties** dialog box presents text boxes for **Name**, **Config Version**, and two **Fence Daemon Properties** parameters: **Post-Join Delay** and **Post-Fail Delay**.

3. At the **Name** text box, specify a name for the cluster. The name should be descriptive enough to distinguish it from other clusters and systems on your network (for example, **`nfs_cluster`** or **`httpd_cluster`**). The cluster name cannot exceed 15 characters.

   > **Tip**
   >
   > Choose the cluster name carefully. The only way to change the name of a Red Hat cluster is to create a new cluster configuration with the new name.

4. (Optional) The **Config Version** value is set to **`1`** by default and is automatically incremented each time you save your cluster configuration. However, if you need to set it to another value, you can specify it at the **Config Version** text box.
5. Specify the **Fence Daemon Properties** parameters: **Post-Join Delay** and **Post-Fail Delay**.
   a. The **Post-Join Delay** parameter is the number of seconds the fence daemon (`fenced`) waits before fencing a node after the node joins the fence domain. The **Post-Join Delay** default value is **`3`**. A typical setting for **Post-Join Delay** is between 20 and 30 seconds, but can vary according to cluster and network performance.

   b. The **Post-Fail Delay** parameter is the number of seconds the fence daemon (`fenced`) waits before fencing a node (a member of the fence domain) after the node has failed. The **Post-Fail Delay** default value is **`0`**. Its value may be varied to suit cluster and network performance.

   > **Note**
   >
   > For more information about **Post-Join Delay** and **Post-Fail Delay**, refer to the fenced(8) man page.

6. Save cluster configuration changes by selecting **File** => **Save**.

3.6. Configuring Fence Devices

Configuring fence devices for the cluster consists of selecting one or more fence devices and specifying fence-device-dependent parameters (for example, name, IP address, login, and password).

To configure fence devices, follow these steps:

1. Click **Fence Devices**. At the bottom of the right frame (labeled **Properties**), click the **Add a Fence Device** button. Clicking **Add a Fence Device** causes the **Fence Device Configuration** dialog box to be displayed (refer to Figure 3-6).

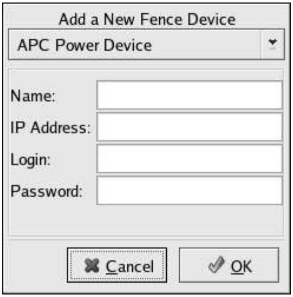


Figure 3-6. Fence Device Configuration

2. At the **Fence Device Configuration** dialog box, click the drop-down box under **Add a New Fence Device** and select the type of fence device to configure.
3. Specify the information in the **Fence Device Configuration** dialog box according to the type of fence device. Refer to the following tables for more information.

Field	Description
Name	A name for the APC device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Table 3-1. Configuring an APC Fence Device

Field	Description
Name	A name for the Brocade device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Table 3-2. Configuring a Brocade Fibre Channel Switch

Field	Description
IP Address	The IP address assigned to the PAP console.
Login	The login name used to access the PAP console.
Password	The password used to authenticate the connection to the PAP console.

Table 3-3. Configuring a Bull Platform Administration Processor (PAP) Interface

Field	Description
Name	The name assigned to the DRAC.
IP Address	The IP address assigned to the DRAC.
Login	The login name used to access the DRAC.
Password	The password used to authenticate the connection to the DRAC.

Table 3-4. Configuring a Dell Remote Access Controller/Modular Chassis (DRAC/MC) Interface

Field	Description
Name	A name for the BladeFrame device connected to the cluster.
CServer	The hostname (and optionally the username in the form of **`username@hostname`**) assigned to the device. Refer to the fence_egenera(8) man page.

Table 3-5. Configuring an Egenera BladeFrame

Field	Description
Name	A name for the GNBD device used to fence the cluster. Note that the GFS server must be accessed via GNBD for cluster node fencing support.
Server	The hostname of each GNBD to disable. For multiple hostnames, separate each hostname with a space.

Table 3-6. Configuring a Global Network Block Device (GNBD) fencing agent

Field	Description
Name	A name for the server with HP iLO support.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.
Hostname	The hostname assigned to the device.

Table 3-7. Configuring an HP Integrated Lights Out (iLO) card

Field	Description
Name	A name for the IBM Bladecenter device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Table 3-8. Configuring an IBM Blade Center that Supports Telnet

Field	Description
Name	A name for the RSA device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Table 3-9. Configuring an IBM Remote Supervisor Adapter II (RSA II)

Field	Description
IP Address	The IP address assigned to the IPMI port.
Login	The login name of a user capable of issuing power on/off commands to the given IPMI port.
Password	The password used to authenticate the connection to the IPMI port.

Table 3-10. Configuring an Intelligent Platform Management Interface (IPMI)

Field	Description
Name	A name to assign the Manual fencing agent. Refer to `fence_manual`(8) for more information.

Table 3-11. Configuring Manual Fencing

Note

Manual fencing is *not* supported for production environments.

Field	Description
Name	A name for the McData device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Table 3-12. Configuring a McData Fibre Channel Switch

Field	Description
Name	A name for the WTI RPS-10 power switch connected to the cluster.
Device	The device the switch is connected to on the controlling host (for example, `/dev/ttys2`).
Port	The switch outlet number.

Table 3-13. Configuring an RPS-10 Power Switch (two-node clusters only)

Field	Description
Name	A name for the SANBox2 device connected to the cluster.
IP Address	The IP address assigned to the device.
Login	The login name used to access the device.
Password	The password used to authenticate the connection to the device.

Table 3-14. Configuring a QLogic SANBox2 Switch

Field	Description
Name	A name for the Vixel switch connected to the cluster.
IP Address	The IP address assigned to the device.
Password	The password used to authenticate the connection to the device.

Table 3-15. Configuring a Vixel SAN Fibre Channel Switch

Field	Description
Name	A name for the WTI power switch connected to the cluster.
IP Address	The IP address assigned to the device.
Password	The password used to authenticate the connection to the device.

Table 3-16. Configuring a WTI Network Power Switch

4. Click **OK**.
5. Choose **File** => **Save** to save the changes to the cluster configuration.

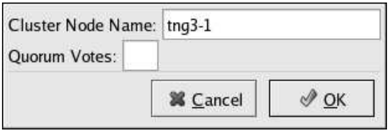
3.7. Adding and Deleting Members

The procedure to add a member to a cluster varies depending on whether the cluster is a newly-configured cluster or a cluster that is already configured and running. To add a member to a new cluster, refer to Section 3.7.1 *Adding a Member to a Cluster*. To add a member to an existing cluster, refer to Section 3.7.2 *Adding a Member to a Running Cluster*. To delete a member from a cluster, refer to Section 3.7.3 *Deleting a Member from a Cluster*.

3.7.1. Adding a Member to a Cluster

To add a member to a new cluster, follow these steps:

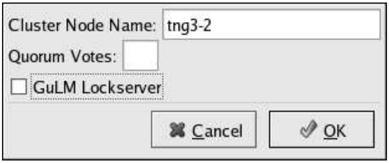
1. Click **Cluster Node**.
2. At the bottom of the right frame (labeled **Properties**), click the **Add a Cluster Node** button. Clicking that button causes a **Node Properties** dialog box to be displayed. For a DLM cluster, the **Node Properties** dialog box presents text boxes for **Cluster Node Name** and **Quorum Votes** (refer to Figure 3-7). For a GULM cluster, the **Node Properties** dialog box presents text boxes for **Cluster Node Name** and **Quorum Votes**, and presents a checkbox for **GULM Lockserver** (refer to Figure 3-8).



Cluster Node Name: tng3-1

Quorum Votes:

Figure 3-7. Adding a Member to a New DLM Cluster



Cluster Node Name: tng3-2

Quorum Votes:

☐ GuLM Lockserver

Figure 3-8. Adding a Member to a New GULM Cluster

3. At the **Cluster Node Name** text box, specify a node name. The entry can be a name or an IP address of the node on the cluster subnet.

Note

Each node must be on the same subnet as the node from which you are running the **Cluster Configuration Tool** and must be defined either in DNS or in the `/etc/hosts` file of each cluster node.

Note

The node on which you are running the **Cluster Configuration Tool** must be explicitly added as a cluster member; the node is not automatically added to the cluster configuration as a result of running the **Cluster Configuration Tool**.

4. Optionally, at the **Quorum Votes** text box, you can specify a value; however in most configurations you can leave it blank. Leaving the **Quorum Votes** text box blank causes the quorum votes value for that node to be set to the default value of `1`.
5. If the cluster is a GULM cluster and you want this node to be a GULM lock server, click the **GULM Lockserver** checkbox (marking it as checked).
6. Click **OK**.
7. Configure fencing for the node:
   a. Click the node that you added in the previous step.
   b. At the bottom of the right frame (below **Properties**), click **Manage Fencing For This Node**. Clicking **Manage Fencing For This Node** causes the **Fence Configuration** dialog box to be displayed.
   c. At the **Fence Configuration** dialog box, bottom of the right frame (below **Properties**), click **Add a New Fence Level**. Clicking **Add a New Fence Level** causes a fence-level element (for example, **Fence-Level-1**, **Fence-Level-2**, and so on) to be displayed below the node in the left frame of the **Fence Configuration** dialog box.
   d. Click the fence-level element.
   e. At the bottom of the right frame (below **Properties**), click **Add a New Fence to this Level**. Clicking **Add a New Fence to this Level** causes the **Fence Properties** dialog box to be displayed.
   f. At the **Fence Properties** dialog box, click the **Fence Device Type** drop-down box and select the fence device for this node. Also, provide additional information required (for example, **Port** and **Switch** for an APC Power Device).
   g. At the **Fence Properties** dialog box, click **OK**. Clicking **OK** causes a fence device element to be displayed below the fence-level element.
   h. To create additional fence devices at this fence level, return to step 6d. Otherwise, proceed to the next step.
   i. To create additional fence levels, return to step 6c. Otherwise, proceed to the next step.
   j. If you have configured all the fence levels and fence devices for this node, click **Close**.

8. Choose **File => Save** to save the changes to the cluster configuration.

3.7.2. Adding a Member to a Running Cluster

The procedure for adding a member to a running cluster depends on whether the cluster contains only two nodes or more than two nodes. To add a member to a running cluster, follow the steps in one of the following sections according to the number of nodes in the cluster:

- For clusters with *only* two nodes —

  Section 3.7.2.1 *Adding a Member to a Running Cluster That Contains Only Two Nodes*
- For clusters with *more than* two nodes —

  Section 3.7.2.2 *Adding a Member to a Running Cluster That Contains More Than Two Nodes*

3.7.2.1. Adding a Member to a Running Cluster That Contains *Only* Two Nodes

To add a member to an existing cluster that is currently in operation, and contains *only* two nodes, follow these steps:

1. Add the node and configure fencing for it as in

   Section 3.7.1 *Adding a Member to a Cluster*.
2. Click **Send to Cluster** to propagate the updated configuration to other running nodes in the cluster.
3. Use the `scp` command to send the updated `/etc/cluster/cluster.conf` file from one of the existing cluster nodes to the new node.
4. At the Red Hat Cluster Suite management GUI **Cluster Status Tool** tab, disable each service listed under **Services**.
5. Stop the cluster software on the two running nodes by running the following commands at each node in this order:

   a. `service rgmanager stop`

   b. `service gfs stop`, if you are using Red Hat GFS

   c. `service clvmd stop`

   d. `service fenced stop`

   e. `service cman stop`

   f. `service ccsd stop`

6. Start cluster software on all cluster nodes (including the added one) by running the following commands in this order:

   a. `service ccsd start`

   b. `service cman start`

   c. `service fenced start`

   d. `service clvmd start`

   e. `service gfs start`, if you are using Red Hat GFS

   f. `service rgmanager start`

7. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

3.7.2.2. Adding a Member to a Running Cluster That Contains *More Than* Two Nodes

To add a member to an existing cluster that is currently in operation, and contains *more than* two nodes, follow these steps:

1. Add the node and configure fencing for it as in

   Section 3.7.1 *Adding a Member to a Cluster*.

2. Click **Send to Cluster** to propagate the updated configuration to other running nodes in the cluster.

3. Use the `scp` command to send the updated `/etc/cluster/cluster.conf` file from one of the existing cluster nodes to the new node.

4. Start cluster services on the new node by running the following commands in this order:

   a. `service ccsd start`

   b. `service lock_gulmd start` or `service cman start` according to the type of lock manager used

   c. `service fenced start` (DLM clusters only)

   d. `service clvmd start`

   e. `service gfs start`, if you are using Red Hat GFS

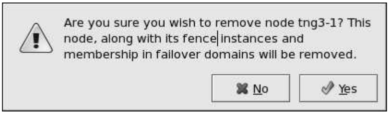
   f. `service rgmanager start`

5. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

3.7.3. Deleting a Member from a Cluster

To delete a member from an existing cluster that is currently in operation, follow these steps:

1. At one of the running nodes (not to be removed), run the Red Hat Cluster Suite management GUI. At the **Cluster Status Tool** tab, under **Services**, disable or relocate each service that is running on the node to be deleted.
2. Stop the cluster software on the node to be deleted by running the following commands at that node in this order:
   a. `service rgmanager stop`
   b. `service gfs stop`, if you are using Red Hat GFS
   c. `service clvmd stop`
   d. `service fenced stop` (DLM clusters only)
   e. `service lock_gulmd stop` or `service cman stop` according to the type of lock manager used
   f. `service ccsd stop`
3. At the **Cluster Configuration Tool** (on one of the running members), delete the member as follows:
   a. If necessary, click the triangle icon to expand the **Cluster Nodes** property.
   b. Select the cluster node to be deleted. At the bottom of the right frame (labeled **Properties**), click the **Delete Node** button.
   c. Clicking the **Delete Node** button causes a warning dialog box to be displayed requesting confirmation of the deletion (Figure 3-9).



Are you sure you wish to remove node tng3-1? This node, along with its fence instances and membership in failover domains will be removed.

Figure 3-9. Confirm Deleting a Member

   d. At that dialog box, click **Yes** to confirm deletion.

   e. Propagate the updated configuration by clicking the **Send to Cluster** button. (Propagating the updated configuration automatically saves the configuration.)

4. Stop the cluster software on the all remaining running nodes (including GULM lock-server nodes for GULM clusters) by running the following commands at each node in this order:

   a. `service rgmanager stop`

   b. `service gfs stop`, if you are using Red Hat GFS

   c. `service clvmd stop`

   d. `service fenced stop` (DLM clusters only)

   e. `service lock_gulmd stop` or `service cman stop` according to the type of lock manager used

   f. `service ccsd stop`

5. Start cluster software on all remaining cluster nodes (including the GULM lock-server nodes for a GULM cluster) by running the following commands in this order:

   a. `service ccsd start`

   b. `service lock_gulmd start` or `service cman start` according to the type of lock manager used

   c. `service fenced start` (DLM clusters only)

   d. `service clvmd start`

   e. `service gfs start`, if you are using Red Hat GFS

   f. `service rgmanager start`

6. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

3.8. Configuring a Failover Domain

A failover domain is a named subset of cluster nodes that are eligible to run a cluster service in the event of a node failure. A failover domain can have the following characteristics:

- Unrestricted — Allows you to specify that a subset of members are preferred, but that a cluster service assigned to this domain can run on any available member.
- Restricted — Allows you to restrict the members that can run a particular cluster service. If none of the members in a restricted failover domain are available, the cluster service cannot be started (either manually or by the cluster software).
- Unordered — When a cluster service is assigned to an unordered failover domain, the member on which the cluster service runs is chosen from the available failover domain members with no priority ordering.
- Ordered — Allows you to specify a preference order among the members of a failover domain. The member at the top of the list is the most preferred, followed by the second member in the list, and so on.

By default, failover domains are unrestricted and unordered.

In a cluster with several members, using a restricted failover domain can minimize the work to set up the cluster to run a cluster service (such as `httpd`), which requires you to set up the configuration identically on all members that run the cluster service). Instead of setting up the entire cluster to run the cluster service, you must set up only the members in the restricted failover domain that you associate with the cluster service.

Tip

> To configure a preferred member, you can create an unrestricted failover domain comprising only one cluster member. Doing that causes a cluster service to run on that cluster member primarily (the preferred member), but allows the cluster service to fail over to any of the other members.

The following sections describe adding a failover domain, removing a failover domain, and removing members from a failover domain:

- Section 3.8.1 *Adding a Failover Domain*
- Section 3.8.2 *Removing a Failover Domain*
- Section 3.8.3 *Removing a Member from a Failover Domain*

3.8.1. Adding a Failover Domain

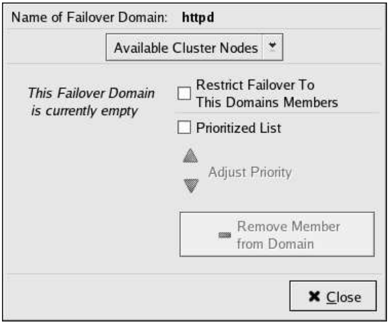
To add a failover domain, follow these steps:

1. At the left frame of the the **Cluster Configuration Tool**, click **Failover Domains**.

2. At the bottom of the right frame (labeled **Properties**), click the **Create a Failover Domain** button. Clicking the **Create a Failover Domain** button causes the **Add Failover Domain** dialog box to be displayed.
3. At the **Add Failover Domain** dialog box, specify a failover domain name at the **Name for new Failover Domain** text box and click **OK**. Clicking **OK** causes the **Failover Domain Configuration** dialog box to be displayed (Figure 3-10).

Note

The name should be descriptive enough to distinguish its purpose relative to other names used in your cluster.

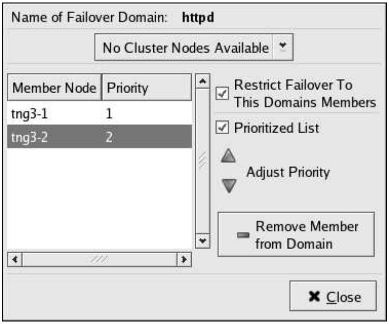


Name of Failover Domain: **httpd**

Available Cluster Nodes
This Failover Domain is currently empty
□ Restrict Failover To This Domains Members
□ Prioritized List
Adjust Priority
Remove Member from Domain
Close

Figure 3-10. Failover Domain Configuration: Configuring a Failover Domain

4. Click the **Available Cluster Nodes** drop-down box and select the members for this failover domain.
5. To restrict failover to members in this failover domain, click (check) the **Restrict Failover To This Domains Members** checkbox. (With **Restrict Failover To This Domains Members** checked, services assigned to this failover domain fail over only to nodes in this failover domain.)
6. To prioritize the order in which the members in the failover domain assume control of a failed cluster service, follow these steps:
   a. Click (check) the **Prioritized List** checkbox (Figure 3-11). Clicking **Prioritized List** causes the **Priority** column to be displayed next to the **Member Node** column.


Name of Failover Domain: **httpd**

Member Node	Priority
tng3-1	1
tng3-2	2

☑ Restrict Failover To This Domains Members

☑ Prioritized List

Figure 3-11. Failover Domain Configuration: Adjusting Priority

b. For each node that requires a priority adjustment, click the node listed in the **Member Node/Priority** columns and adjust priority by clicking one of the **Adjust Priority** arrows. Priority is indicated by the position in the **Member Node** column and the value in the **Priority** column. The node priorities are listed highest to lowest, with the highest priority node at the top of the **Member Node** column (having the lowest **Priority** number).

7. Click **Close** to create the domain.
8. At the **Cluster Configuration Tool**, perform one of the following actions depending on whether the configuration is for a new cluster or for one that is operational and running:
   - New cluster — If this is a new cluster, choose **File => Save** to save the changes to the cluster configuration.
   - Running cluster — If this cluster is operational and running, and you want to propagate the change immediately, click the **Send to Cluster** button. Clicking **Send to Cluster** automatically saves the configuration change. If you do not want to propagate the change immediately, choose **File => Save** to save the changes to the cluster configuration.

3.8.2. Removing a Failover Domain

To remove a failover domain, follow these steps:

1. At the left frame of the the **Cluster Configuration Tool**, click the failover domain that you want to delete (listed under **Failover Domains**).
2. At the bottom of the right frame (labeled **Properties**), click the **Delete Failover Domain** button. Clicking the **Delete Failover Domain** button causes a warning dialog box do be displayed asking if you want to remove the failover domain. Confirm that the failover domain identified in the warning dialog box is the one you want to delete and click **Yes**. Clicking **Yes** causes the failover domain to be removed from the list of failover domains under **Failover Domains** in the left frame of the **Cluster Configuration Tool**.
3. At the **Cluster Configuration Tool**, perform one of the following actions depending on whether the configuration is for a new cluster or for one that is operational and running:
   - New cluster — If this is a new cluster, choose **File => Save** to save the changes to the cluster configuration.
   - Running cluster — If this cluster is operational and running, and you want to propagate the change immediately, click the **Send to Cluster** button. Clicking **Send to Cluster** automatically saves the configuration change. If you do not want to propagate the change immediately, choose **File => Save** to save the changes to the cluster configuration.

3.8.3. Removing a Member from a Failover Domain

To remove a member from a failover domain, follow these steps:

1. At the left frame of the the **Cluster Configuration Tool**, click the failover domain that you want to change (listed under **Failover Domains**).
2. At the bottom of the right frame (labeled **Properties**), click the **Edit Failover Domain Properties** button. Clicking the **Edit Failover Domain Properties** button causes the **Failover Domain Configuration** dialog box to be displayed (Figure 3-10).
3. At the **Failover Domain Configuration** dialog box, in the **Member Node** column, click the node name that you want to delete from the failover domain and click the **Remove Member from Domain** button. Clicking **Remove Member from Domain** removes the node from the **Member Node** column. Repeat this step for each node that is to be deleted from the failover domain. (Nodes must be deleted one at a time.)
4. When finished, click **Close**.

5. At the **Cluster Configuration Tool**, perform one of the following actions depending on whether the configuration is for a new cluster or for one that is operational and running:
   - New cluster — If this is a new cluster, choose **File => Save** to save the changes to the cluster configuration.
   - Running cluster — If this cluster is operational and running, and you want to propagate the change immediately, click the **Send to Cluster** button. Clicking **Send to Cluster** automatically saves the configuration change. If you do not want to propagate the change immediately, choose **File => Save** to save the changes to the cluster configuration.

3.9. Adding Cluster Resources

To specify a device for a cluster service, follow these steps:

1. On the **Resources** property of the **Cluster Configuration Tool**, click the **Create a Resource** button. Clicking the **Create a Resource** button causes the **Resource Configuration** dialog box to be displayed.
2. At the **Resource Configuration** dialog box, under **Select a Resource Type**, click the drop-down box. At the drop-down box, select a resource to configure. The resource options are described as follows:

   GFS

   > **Name** — Create a name for the file system resource.
   >
   > **Mount Point** — Choose the path to which the file system resource is mounted.
   >
   > **Device** — Specify the device file associated with the file system resource.
   >
   > **Options** — Options to pass to the `mkfs` call for the new file system.
   >
   > **File System ID** — When creating a new file system resource, you can leave this field blank. Leaving the field blank causes a file system ID to be assigned automatically after you click **OK** at the **Resource Configuration** dialog box. If you need to assign a file system ID explicitly, specify it in this field.
   >
   > **Force Unmount** checkbox — If checked, forces the file system to unmount. The default setting is unchecked.

   File System

   > **Name** — Create a name for the file system resource.

File System Type — Choose the file system for the resource using the drop-down menu.

Mount Point — Choose the path to which the file system resource is mounted.

Device — Specify the device file associated with the file system resource.

Options — Options to pass to the `mkfs` call for the new file system.

File System ID — When creating a new file system resource, you can leave this field blank. Leaving the field blank causes a file system ID to be assigned automatically after you click **OK** at the **Resource Configuration** dialog box. If you need to assign a file system ID explicitly, specify it in this field.

Checkboxes — Specify mount and unmount actions when a service is stopped (for example, when disabling or relocating a service):

- **Force unmount** — If checked, forces the file system to unmount. The default setting is unchecked.
- **Reboot host node if unmount fails** — If checked, reboots the node if unmounting this file system fails. The default setting is unchecked.
- **Check file system before mounting** — If checked, causes `fsck` to be run on the file system before mounting it. The default setting is unchecked.

IP Address

IP Address — Type the IP address for the resource.

Monitor Link checkbox — Check the box to enable or disable link status monitoring of the IP address resource

NFS Mount

Name — Create a symobolic name for the NFS mount.

Mount Point — Choose the path to which the file system resource is mounted.

Host — Specify the NFS server name.

Export Path — NFS export on the server.

NFS and **NFS4** options — Specify NFS protocol:

- **NFS** — Specifies using NFSv3 protocol. The default setting is **NFS**.
- **NFS4** — Specifies using NFSv4 protocol.

Options — NFS-specific options to pass to the `mkfs` call for the new file system. For more information, refer to the nfs(5) man page.

Force Unmount checkbox — If checked, forces the file system to unmount. The default setting is unchecked.

NFS Client

Name — Enter a name for the NFS client resource.

Target — Enter a target for the NFS client resource. Supported targets are hostnames, IP addresses (with wild-card support), and netgroups.

Read-Write and **Read Only** options — Specify the type of access rights for this NFS client resource:

- **Read-Write** — Specifies that the NFS client has read-write access. The default setting is **Read-Write**.
- **Read Only** — Specifies that the NFS client has read-only access.

Options — Additional client access rights. For more information, refer to the exports(5) man page, General Options

NFS Export

Name — Enter a name for the NFS export resource.

Script

Name — Enter a name for the custom user script.

File (with path) — Enter the path where this custom script is located (for example, `/etc/init.d/`*`userscript`*)

Samba Service

Name — Enter a name for the Samba server.

Work Group — Enter the Windows workgroup name or Windows NT domain of the Samba service.

Note

When creating or editing a cluster service, connect a Samba-service resource directly to the service, *not* to a resource within a service. That is, at the **Service Management** dialog box, use either **Create a new resource for this service** or **Add a Shared Resource to this service**; do *not* use **Attach a new Private Resource to the Selection** or **Attach a Shared Resource to the selection**.

3. When finished, click **OK**.
4. Choose **File => Save** to save the change to the `/etc/cluster/cluster.conf` configuration file.

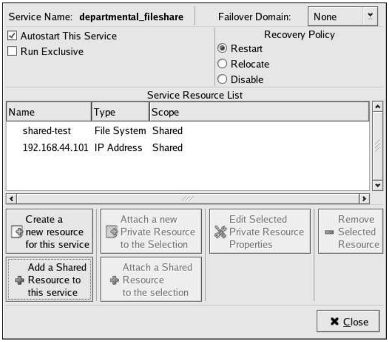
3.10. Adding a Cluster Service to the Cluster

To add a cluster service to the cluster, follow these steps:

1. At the left frame, click **Services**.
2. At the bottom of the right frame (labeled **Properties**), click the **Create a Service** button. Clicking **Create a Service** causes the **Add a Service** dialog box to be displayed.
3. At the **Add a Service** dialog box, type the name of the service in the **Name** text box and click **OK**. Clicking **OK** causes the **Service Management** dialog box to be displayed (refer to Figure 3-12).

   Tip

   Use a descriptive name that clearly distinguishes the service from other services in the cluster.



Service Name: **departmental_fileshare**

Failover Domain: None

☑ Autostart This Service

□ Run Exclusive

Recovery Policy

● Restart
○ Relocate
○ Disable

Service Resource List

Name	Type	Scope
shared-test	File System	Shared
192.168.44.101	IP Address	Shared

Create a new resource for this service
Attach a new Private Resource to the Selection
Edit Selected Private Resource Properties
Remove Selected Resource
Add a Shared Resource to this service
Attach a Shared Resource to the selection
Close

Figure 3-12. Adding a Cluster Service

4. If you want to restrict the members on which this cluster service is able to run, choose a failover domain from the **Failover Domain** drop-down box. (Refer to Section 3.8 *Configuring a Failover Domain* for instructions on how to configure a failover domain.)
5. **Autostart This Service** checkbox — This is checked by default. If **Autostart This Service** is checked, the service is started automatically when a cluster is started and running. If **Autostart This Service** is *not* checked, the service must be started manually any time the cluster comes up from stopped state.

6. **Run Exclusive** checkbox — This sets a policy wherein the service only runs on nodes that have *no other* services running on them. For example, for a very busy web server that is clustered for high availability, it would would be advisable to keep that service on a node alone with no other services competing for his resources — that is, **Run Exclusive** checked. On the other hand, services that consume few resources (like NFS and Samba), can run together on the same node without little concern over contention for resources. For those types of services you can leave the **Run Exclusive** unchecked.

7. Select a recovery policy to specify how the resource manager should recover from a service failure. At the upper right of the **Service Management** dialog box, there are three **Recovery Policy** options available:

   - **Restart** — Restart the service in the node the service is currently located. The default setting is **Restart**. If the service cannot be restarted in the the current node, the service is relocated.

   - **Relocate** — Relocate the service before restarting. Do not restart the node where the service is currently located.

   - **Disable** — Do not restart the service at all.

8. Click the **Add a Shared Resource to this service** button and choose the a resource listed that you have configured in Section 3.9 *Adding Cluster Resources*.

   > **Note**
   >
   > If you are adding a Samba-service resource, connect a Samba-service resource directly to the service, *not* to a resource within a service. That is, at the **Service Management** dialog box, use either **Create a new resource for this service** or **Add a Shared Resource to this service**; do *not* use **Attach a new Private Resource to the Selection** or **Attach a Shared Resource to the selection**.

9. If needed, you may also create a *private* resource that you can create that becomes a subordinate resource by clicking on the **Attach a new Private Resource to the Selection** button. The process is the same as creating a shared resource described in Section 3.9 *Adding Cluster Resources*. The private resource will appear as a child to the shared resource to which you associated with the shared resource. Click the triangle icon next to the shared resource to display any private resources associated.

10. When finished, click **OK**.

11. Choose **File => Save** to save the changes to the cluster configuration.

Note

To verify the existence of the IP service resource used in a cluster service, you must use the `/sbin/ip addr list` command on a cluster node. The following output shows the `/sbin/ip addr list` command executed on a node running a cluster service:

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
       valid_lft forever preferred_lft forever
```

3.11. Propagating The Configuration File: New Cluster

For newly defined clusters, you must propagate the configuration file to the cluster nodes as follows:

1. Log in to the node where you created the configuration file.
2. Using the `scp` command, copy the `/etc/cluster/cluster.conf` file to all nodes in the cluster.

   Note

   Propagating the cluster configuration file this way is necessary for the first time a cluster is created. Once a cluster is installed and running, the cluster configuration file is propagated using the Red Hat cluster management GUI **Send to Cluster** button. For more information about propagating the cluster configuration using the GUI **Send to Cluster** button, refer to Section 4.4 *Modifying the Cluster Configuration*.

3.12. Starting the Cluster Software

After you have propagated the cluster configuration to the cluster nodes you can either reboot each node or start the cluster software on each cluster node by running the following commands at each node in this order:

1. `service ccsd start`
2. `service lock_gulmd start` or `service cman start` according to the type of lock manager used
3. `service fenced start` (DLM clusters only)
4. `service clvmd start`
5. `service gfs start`, if you are using Red Hat GFS
6. `service rgmanager start`
7. Start the Red Hat Cluster Suite management GUI. At the **Cluster Configuration Tool** tab, verify that the configuration is correct. At the **Cluster Status Tool** tab verify that the nodes and services are running as expected.

Chapter 4.

Cluster Administration

This chapter describes the various administrative tasks for maintaining a cluster after it has been installed and configured.

4.1. Overview of the Cluster Status Tool

The **Cluster Status Tool** is part of the Red Hat Cluster Suite management GUI, (the `system-config-cluster` package) and is accessed by a tab in the Red Hat Cluster Suite management GUI. The **Cluster Status Tool** displays the status of cluster members and services and provides control of cluster services.

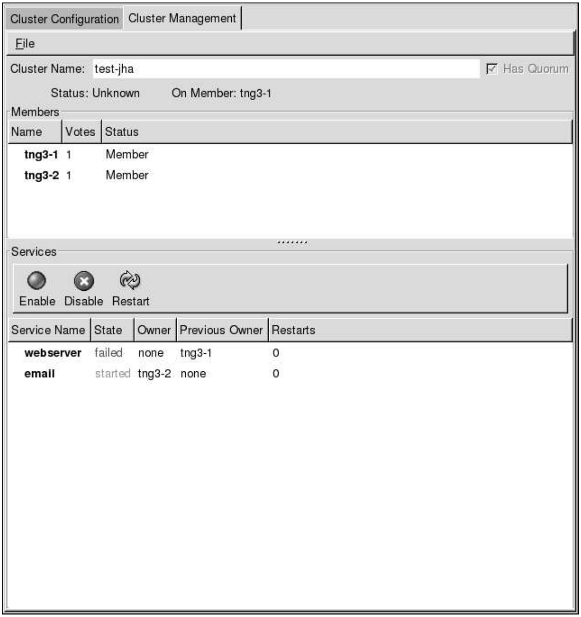
The members and services displayed in the **Cluster Status Tool** are determined by the cluster configuration file (`/etc/cluster/cluster.conf`). The cluster configuration file is maintained via the **Cluster Configuration Tool** in the cluster management GUI.

Warning

> Do not manually edit the contents of the `/etc/cluster/cluster.conf` file without guidance from an authorized Red Hat representative or unless you fully understand the consequences of editing the `/etc/cluster/cluster.conf` file manually.

You can access the **Cluster Status Tool** by clicking the **Cluster Management** tab at the cluster management GUI to Figure 4-1).

Use the **Cluster Status Tool** to enable, disable, restart, or relocate a service. To enable a service, select the service in the **Services** area and click **Enable**. To disable a service, select the service in the **Services** area and click **Disable**. To restart a service, select the service in the **Services** area and click **Restart**. To relocate service from one member to another, drag the service to another member and drop the service onto that member. Relocating a member restarts the service on that member. (Relocating a service to its current member — that is, dragging a service to its current member and dropping the service onto that member — restarts the service.)



Cluster Name: test-jha ☑ Has Quorum

Status: Unknown On Member: tng3-1

Members

Name	Votes	Status
tng3-1	1	Member
tng3-2	1	Member

Services

Service Name	State	Owner	Previous Owner	Restarts
webserver	failed	none	tng3-1	0
email	started	tng3-2	none	0

Figure 4-1. Cluster Status Tool

4.2. Displaying Cluster and Service Status

Monitoring cluster and application service status can help identify and resolve problems in the cluster environment. The following tools assist in displaying cluster status information:

- The **Cluster Status Tool**
- The `clustat` utility

Important

Members that are not running the cluster software cannot determine or report the status of other members of the cluster.

Cluster and service status includes the following information:

- Cluster member system status
- Service status and which cluster system is running the service or owns the service

The following tables describe how to analyze the status information shown by the **Cluster Status Tool** and the `clustat` utility.

Member Status	Description
Member	The node is part of the cluster. Note: A node can be a member of a cluster; however, the node may be inactive and incapable of running services. For example, if `rgmanager` is not running on the node, but all other cluster software components are running in the node, the node appears as a **Member** in the **Cluster Status Tool**. However, without `rgmanager` running, the node does not appear in the `clustat` display.
Dead	The member system is unable to participate as a cluster member. The most basic cluster software is not running on the node.

Table 4-1. Member Status for the Cluster Status Tool

Member Status	Description
Online	The node is communicating with other nodes in the cluster.
Inactive	The node is unable to communicate with the other nodes in the cluster. If the node is inactive, `clustat` does not display the node. If `rgmanager` is not running in a node, the node is inactive. Note: Although a node is inactive, it may still appear as a **Member** in the **Cluster Status Tool**. However, if the node is inactive, it is incapable of running services.

Table 4-2. Member Status for `clustat`

Service Status	Description
Started	The service resources are configured and available on the cluster system that owns the service.
Pending	The service has failed on a member and is pending start on another member.
Disabled	The service has been disabled, and does not have an assigned owner. A disabled service is never restarted automatically by the cluster.
Stopped	The service is not running; it is waiting for a member capable of starting the service. A service remains in the stopped state if autostart is disabled.
Failed	The service has failed to start on the cluster and cannot successfully stop the service. A failed service is never restarted automatically by the cluster.

Table 4-3. Service Status

The **Cluster Status Tool** displays the current cluster status in the **Services** area and automatically updates the status every 10 seconds. Additionally, you can display a snapshot of the current cluster status from a shell prompt by invoking the `clustat` utility. Example 4-1 shows the output of the `clustat` utility.

```
# clustat
Member Status: Quorate, Group Member

  Member Name                               State      ID
  ------ ----                               -----      --
  tng3-2                                    Online     0x0000000000000002
  tng3-1                                    Online     0x0000000000000001

  Service Name         Owner (Last)                    State
  --------- -----      ----- ------                    -----
  webserver            (tng3-1                       ) failed
  email                tng3-2                          started
```

Example 4-1. Output of `clustat`

To monitor the cluster and display status at specific time intervals from a shell prompt, invoke `clustat` with the `-i` *`time`* option, where *`time`* specifies the number of seconds between status snapshots. The following example causes the `clustat` utility to display cluster status every 10 seconds:

```
#clustat -i 10
```

4.3. Starting and Stopping the Cluster Software

To start the cluster software on a member, type the following commands in this order:

1. `service ccsd start`
2. `service lock_gulmd start` or `service cman start` according to the type of lock manager used
3. `service fenced start` (DLM clusters only)
4. `service clvmd start`
5. `service gfs start`, if you are using Red Hat GFS
6. `service rgmanager start`

To stop the cluster software on a member, type the following commands in this order:

1. `service rgmanager stop`
2. `service gfs stop`, if you are using Red Hat GFS
3. `service clvmd stop`
4. `service fenced stop` (DLM clusters only)
5. `service lock_gulmd stop` or `service cman stop` according to the type of lock manager used
6. `service ccsd stop`

Stopping the cluster services on a member causes its services to fail over to an active member.

4.4. Modifying the Cluster Configuration

To modify the cluster configuration (the cluster configuration file (`/etc/cluster/cluster.conf`), use the **Cluster Configuration Tool**. For more information about using the **Cluster Configuration Tool**, refer to Chapter 3 *Installing and Configuring Red Hat Cluster Suite Software*.

Warning

Do not manually edit the contents of the `/etc/cluster/cluster.conf` file without guidance from an authorized Red Hat representative or unless you fully understand the consequences of editing the `/etc/cluster/cluster.conf` file manually.

Important

Although the **Cluster Configuration Tool** provides a **Quorum Votes** parameter in the **Properties** dialog box of each cluster member, that parameter is intended *only* for use during initial cluster configuration. Furthermore, it is recommended that you retain the default **Quorum Votes** value of `1`. For more information about using the **Cluster Configuration Tool**, refer to Chapter 3 *Installing and Configuring Red Hat Cluster Suite Software*.

To edit the cluster configuration file, click the **Cluster Configuration** tab in the cluster configuration GUI. Clicking the **Cluster Configuration** tab displays a graphical representation of the cluster configuration. Change the configuration file according the the following steps:

1. Make changes to cluster elements (for example, create a service).
2. Propagate the updated configuration file throughout the cluster by clicking **Send to Cluster**.

   Note

   The **Cluster Configuration Tool** does not display the **Send to Cluster** button if the cluster is new and has not been started yet, or if the node from which you are running the **Cluster Configuration Tool** is not a member of the cluster. If the **Send to Cluster** button is not displayed, you can still use the **Cluster Configuration Tool**; however, you cannot propagate the configuration. You can still *save* the configuration file. For information about using the **Cluster Configuration Tool** for a new cluster configuration, refer to Chapter 3 *Installing and Configuring Red Hat Cluster Suite Software*.

3. Clicking **Send to Cluster** causes a **Warning** dialog box to be displayed. Click **Yes** to save and propagate the configuration.
4. Clicking **Yes** causes an **Information** dialog box to be displayed, confirming that the current configuration has been propagated to the cluster. Click **OK**.
5. Click the **Cluster Management** tab and verify that the changes have been propagated to the cluster members.

4.5. Backing Up and Restoring the Cluster Database

The **Cluster Configuration Tool** automatically retains backup copies of the three most recently used configuration files (besides the currently used configuration file). Retaining the backup copies is useful if the cluster does not function correctly because of misconfiguration and you need to return to a previous working configuration.

Each time you save a configuration file, the **Cluster Configuration Tool** saves backup copies of the three most recently used configuration files as `/etc/cluster/cluster.conf.bak.1`, `/etc/cluster/cluster.conf.bak.2`, and `/etc/cluster/cluster.conf.bak.3`. The backup file `/etc/cluster/cluster.conf.bak.1` is the newest backup, `/etc/cluster/cluster.conf.bak.2` is the second newest backup, and `/etc/cluster/cluster.conf.bak.3` is the third newest backup.

If a cluster member becomes inoperable because of misconfiguration, restore the configuration file according to the following steps:

1. At the **Cluster Configuration Tool** tab of the Red Hat Cluster Suite management GUI, click **File => Open**.
2. Clicking **File => Open** causes the **system-config-cluster** dialog box to be displayed.
3. At the the **system-config-cluster** dialog box, select a backup file (for example, `/etc/cluster/cluster.conf.bak.1`). Verify the file selection in the **Selection** box and click **OK**.
4. Increment the configuration version beyond the current working version number as follows:
    a. Click **Cluster => Edit Cluster Properties**.
    b. At the **Cluster Properties** dialog box, change the **Config Version** value and click **OK**.
5. Click **File => Save As**.
6. Clicking **File => Save As** causes the **system-config-cluster** dialog box to be displayed.
7. At the the **system-config-cluster** dialog box, select `/etc/cluster/cluster.conf` and click **OK**. (Verify the file selection in the **Selection** box.)
8. Clicking **OK** causes an **Information** dialog box to be displayed. At that dialog box, click **OK**.
9. Propagate the updated configuration file throughout the cluster by clicking **Send to Cluster**.

> **Note**
>
> The **Cluster Configuration Tool** does not display the **Send to Cluster** button if the cluster is new and has not been started yet, or if the node from which you are running the **Cluster Configuration Tool** is not a member of the cluster. If the **Send to Cluster** button is not displayed, you can still use the **Cluster Configuration Tool**; however, you cannot propagate the configuration. You can still *save* the configuration file. For information about using the **Cluster Configuration Tool** for a new cluster configuration, refer to Chapter 3 *Installing and Configuring Red Hat Cluster Suite Software*.

10. Clicking **Send to Cluster** causes a **Warning** dialog box to be displayed. Click **Yes** to propagate the configuration.
11. Click the **Cluster Management** tab and verify that the changes have been propagated to the cluster members.

4.6. Updating the Cluster Software

For information about updating the cluster software, contact an authorized Red Hat support representative.

4.7. Changing the Cluster Name

Although the **Cluster Configuration Tool** provides a **Cluster Properties** dialog box with a cluster **Name** parameter, the parameter is intended *only* for use during initial cluster configuration. The only way to change the name of a Red Hat cluster is to create a new cluster with the new name. For more information about using the **Cluster Configuration Tool**, refer to Chapter 3 *Installing and Configuring Red Hat Cluster Suite Software*.

4.8. Disabling the Cluster Software

It may become necessary to temporarily disable the cluster software on a cluster member. For example, if a cluster member experiences a hardware failure, you may want to reboot that member, but prevent it from rejoining the cluster to perform maintenance on the system.

Use the `/sbin/chkconfig` command to stop the member from joining the cluster at bootup as follows:

```
chkconfig --level 2345 rgmanager off
chkconfig --level 2345 gfs off
chkconfig --level 2345 clvmd off
chkconfig --level 2345 fenced off
chkconfig --level 2345 lock_gulmd off
chkconfig --level 2345 cman off
chkconfig --level 2345 ccsd off
```

Once the problems with the disabled cluster member have been resolved, use the following commands to allow the member to rejoin the cluster:

```
chkconfig --level 2345 rgmanager on
chkconfig --level 2345 gfs on
chkconfig --level 2345 clvmd on
chkconfig --level 2345 fenced on
chkconfig --level 2345 lock_gulmd on
chkconfig --level 2345 cman on
chkconfig --level 2345 ccsd on
```

You can then reboot the member for the changes to take effect or run the following commands in the order shown to restart cluster software:

1. `service ccsd start`
2. `service lock_gulmd start` or `service cman start` according to the type of lock manager used
3. `service fenced start` (DLM clusters only)
4. `service clvmd start`
5. `service gfs start`, if you are using Red Hat GFS
6. `service rgmanager start`

4.9. Diagnosing and Correcting Problems in a Cluster

For information about diagnosing and correcting problems in a cluster, contact an authorized Red Hat support representative.

Chapter 5.

Setting Up Apache HTTP Server

This chapter contains instructions for configuring Red Hat Enterprise Linux to make the Apache HTTP Server highly available.

The following is an example of setting up a cluster service that fails over an Apache HTTP Server. Although the actual variables used in the service depend on the specific configuration, the example may assist in setting up a service for a particular environment.

5.1. Apache HTTP Server Setup Overview

First, configure Apache HTTP Server on all nodes in the cluster. If using a failover domain , assign the service to all cluster nodes configured to run the Apache HTTP Server. Refer to Section 3.8 *Configuring a Failover Domain* for instructions. The cluster software ensures that only one cluster system runs the Apache HTTP Server at one time. The example configuration consists of installing the `httpd` RPM package on all cluster nodes (or on nodes in the failover domain, if used) and configuring a shared GFS shared resource for the Web content.

When installing the Apache HTTP Server on the cluster systems, run the following command to ensure that the cluster nodes do not automatically start the service when the system boots:

```
chkconfig --del httpd
```

Rather than having the system init scripts spawn the `httpd` daemon, the cluster infrastructure initializes the service on the active cluster node. This ensures that the corresponding IP address and file system mounts are active on only one cluster node at a time.

When adding an `httpd` service, a *floating* IP address must be assigned to the service so that the IP address will transfer from one cluster node to another in the event of failover or service relocation. The cluster infrastructure binds this IP address to the network interface on the cluster system that is currently running the Apache HTTP Server. This IP address ensures that the cluster node running `httpd` is transparent to the clients accessing the service.

The file systems that contain the Web content cannot be automatically mounted on the shared storage resource when the cluster nodes boot. Instead, the cluster software must mount and unmount the file system as the `httpd` service is started and stopped. This prevents the cluster systems from accessing the same data simultaneously, which may result in data corruption. Therefore, do not include the file systems in the `/etc/fstab` file.

5.2. Configuring Shared Storage

To set up the shared file system resource, perform the following tasks as root on one cluster system:

1. On one cluster node, use the interactive `parted` utility to create a partition to use for the document root directory. Note that it is possible to create multiple document root directories on different disk partitions. Refer to Section 2.5.3.1 *Partitioning Disks* for more information.
2. Use the `mkfs` command to create an ext3 file system on the partition you created in the previous step. Specify the drive letter and the partition number. For example:
   ```
   mkfs -t ext3 /dev/sde3
   ```
3. Mount the file system that contains the document root directory. For example:
   ```
   mount /dev/sde3 /var/www/html
   ```
   Do not add this mount information to the `/etc/fstab` file because only the cluster software can mount and unmount file systems used in a service.
4. Copy all the required files to the document root directory.
5. If you have CGI files or other files that must be in different directories or in separate partitions, repeat these steps, as needed.

5.3. Installing and Configuring the Apache HTTP Server

The Apache HTTP Server must be installed and configured on all nodes in the assigned failover domain, if used, or in the cluster. The basic server configuration must be the same on all nodes on which it runs for the service to fail over correctly. The following example shows a basic Apache HTTP Server installation that includes no third-party modules or performance tuning.

On all node in the cluster (or nodes in the failover domain, if used), install the `httpd` RPM package. For example:

```
rpm -Uvh httpd-<version>.<arch>.rpm
```

To configure the Apache HTTP Server as a cluster service, perform the following tasks:

1. Edit the `/etc/httpd/conf/httpd.conf` configuration file and customize the file according to your configuration. For example:
   - Specify the directory that contains the HTML files. Also specify this mount point when adding the service to the cluster configuration. It is only required to change this field if the mountpoint for the website's content differs from the default setting of `/var/www/html/`. For example:
     ```
     DocumentRoot "/mnt/httpdservice/html"
     ```

- Specify a unique IP address to which the service will listen for requests. For example:

  ```
  Listen 192.168.1.100:80
  ```

  This IP address then must be configured as a cluster resource for the service using the **Cluster Configuration Tool**.

- If the script directory resides in a non-standard location, specify the directory that contains the CGI programs. For example:

  ```
  ScriptAlias /cgi-bin/ "/mnt/httpdservice/cgi-bin/"
  ```

- Specify the path that was used in the previous step, and set the access permissions to default to that directory. For example:

  ```
  <Directory /mnt/httpdservice/cgi-bin">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
  </Directory>
  ```

  Additional changes may need to be made to tune the Apache HTTP Server or add module functionality. For information on setting up other options, refer to the *Red Hat Enterprise Linux System Administration Guide* and the *Red Hat Enterprise Linux Reference Guide*.

2. The standard Apache HTTP Server start script, `/etc/rc.d/init.d/httpd` is also used within the cluster framework to start and stop the Apache HTTP Server on the active cluster node. Accordingly, when configuring the service, specify this script by adding it as a **Script** resource in the **Cluster Configuration Tool**.
3. Copy the configuration file over to the other nodes of the cluster (or nodes of the failover domain, if configured).

Before the service is added to the cluster configuration, ensure that the Apache HTTP Server directories are not mounted. Then, on one node, invoke the **Cluster Configuration Tool** to add the service, as follows. This example assumes a failover domain named `httpd-domain` was created for this service.

1. Add the init script for the Apache HTTP Server service.
   - Select the **Resources** tab and click **Create a Resource**. The **Resources Configureation** properties dialog box is displayed.
   - Select **Script** form the drop down menu.
   - Enter a **Name** to be associated with the Apache HTTP Server service.
   - Specify the path to the Apache HTTP Server init script (for example, **`/etc/rc.d/init.d/httpd`**) in the **File (with path)** field.
   - Click **OK**.

2. Add a device for the Apache HTTP Server content files and/or custom scripts.
   - Click **Create a Resource**.
   - In the **Resource Configuration** dialog, select **File System** from the drop-down menu.
   - Enter the **Name** for the resource (for example, `httpd-content`.
   - Choose **ext3** from the **File System Type** drop-down menu.
   - Enter the mount point in the **Mount Point** field (for example, `/var/www/html/`).
   - Enter the device special file name in the **Device** field (for example, `/dev/sda3`).

3. Add an IP address for the Apache HTTP Server service.
   - Click **Create a Resource**.
   - Choose **IP Address** from the drop-down menu.
   - Enter the **IP Address** to be associatged with the Apache HTTP Server service.
   - Make sure that the **Monitor Link** checkbox is left checked.
   - Click **OK**.

4. Click the **Services** property.
5. Create the Apache HTTP Server service.
   - Click **Create a Service**. Type a **Name** for the service in the **Add a Service** dialog.
   - In the **Service Management** dialog, select a **Failover Domain** from the drop-down menu or leave it as **None**.
   - Click the **Add a Shared Resource to this service** button. From the available list, choose each resource that you created in the previous steps. Repeat this step until all resources have been added.
   - Click **OK**.

6. Choose **File => Save** to save your changes.

II. Configuring a Linux Virtual Server Cluster

Building a Linux Virtual Server (LVS) system offers highly-available and scalable solution for production services using specialized routing and load-balancing techniques configured through the **Piranha Configuration Tool**. This part discusses the configuration of high-performance systems and services with Red Hat Enterprise Linux and LVS.

Table of Contents

Chapter 6.

Introduction to Linux Virtual Server

Using Red Hat Enterprise Linux, it is possible to create highly available server clustering solutions able to withstand many common hardware and software failures with little or no interruption of critical services. By allowing multiple computers to work together in offering these critical services, system administrators can plan and execute system maintenance and upgrades without service interruption.

The chapters in this part guide you through the following steps in understanding and deploying a clustering solution based on the Red Hat Enterprise Linux *Linux Virtual Server* (LVS) technology:

- Explains the Linux Virtual Server technology used by Red Hat Enterprise Linux to create a load-balancing cluster
- Explains how to configure a Red Hat Enterprise Linux LVS cluster
- Guides you through the **Piranha Configuration Tool**, a graphical interface used for configuring and monitoring an LVS cluster

6.1. Technology Overview

Red Hat Enterprise Linux implements highly available server solutions via clustering. It is important to note that *cluster* computing consists of three distinct branches:

- *Compute clustering* (such as Beowulf) uses multiple machines to provide greater computing power for computationally intensive tasks. This type of clustering is not addressed by Red Hat Enterprise Linux.
- *High-availability (HA) clustering* uses multiple machines to add an extra level of reliability for a service or group of services.
- *Load-balance clustering* uses specialized routing techniques to dispatch traffic to a pool of servers.

Red Hat Enterprise Linux addresses the latter two types of clustering technology. Using a collection of programs to monitor the health of the systems and services in the cluster.

Note

The clustering technology included in Red Hat Enterprise Linux is not synonymous with *fault tolerance*. Fault tolerant systems use highly specialized and often very expensive hardware to implement a fully redundant environment in which services can run uninterrupted by hardware failures.

However, fault tolerant systems do not account for operator and software errors which Red Hat Enterprise Linux can address through service redundancy. Also, since Red Hat Enterprise Linux is designed to run on commodity hardware, it creates an environment with a high level of system availability at a fraction of the cost of fault tolerant hardware.

6.2. Basic Configurations

While Red Hat Enterprise Linux can be configured in a variety of different ways, the configurations can be broken into two major categories:

- High-availability clusters using Red Hat Cluster Manager
- Load-balancing clusters using Linux Virtual Servers

This part explains what a load-balancing cluster system is and how to configure a load-balancing system using *Linux Virtual Servers* on Red Hat Enterprise Linux.

6.2.1. Load-Balancing Clusters Using Linux Virtual Servers

To an outside user accessing a hosted service (such as a website or database application), a Linux Virtual Server (LVS) cluster appears as one server. In reality, however, the user is actually accessing a cluster of two or more servers behind a pair of redundant LVS routers that distribute client requests evenly throughout the cluster system. Load-balanced clustered services allow administrators to use commodity hardware and Red Hat Enterprise Linux to create continuous and consistent access to all hosted services while also addressing availability requirements.

An LVS cluster consists of at least two layers. The first layer is composed of a pair of similarly configured Linux machines or *cluster members*. One of these machine acts as the *LVS routers*, configured to direct requests from the Internet to the cluster. The second layer consists of a cluster of machines called *real servers*. The real servers provide the critical services to the end-user while the LVS router balances the load on these servers.

For a detailed overview of LVS clustering, refer to Chapter 7 *Linux Virtual Server Overview*.

Chapter 7.

Linux Virtual Server Overview

Red Hat Enterprise Linux LVS clustering uses a Linux machine called the *active router* to send requests from the Internet to a pool of servers. To accomplish this, LVS clusters consist of two basic machine classifications — the LVS routers (one active and one backup) and a pool of real servers which provide the critical services.

The active router serves two roles in the cluster:

- To balance the load on the real servers.
- To check the integrity of the services on each of the real servers.

The backup router's job is to monitor the active router and assume its role in the event of failure.

7.1. A Basic LVS Configuration

Figure 7-1 shows a simple LVS cluster consisting of two layers. On the first layer are two LVS routers — one active and one backup. Each of the LVS routers has two network interfaces, one interface on the Internet and one on the private network, enabling them to regulate traffic between the two networks. For this example the active router is using *Network Address Translation* or *NAT* to direct traffic from the Internet to a variable number of real servers on the second layer, which in turn provide the necessary services. Therefore, the real servers in this example are connected to a dedicated private network segment and pass all public traffic back and forth through the active LVS router. To the outside world, the server cluster appears as one entity.

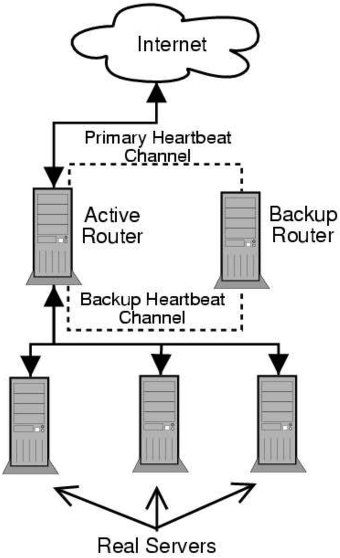


Figure 7-1. A Basic LVS Configuration

Service requests arriving at the LVS cluster are addressed to a *virtual IP* address or VIP. This is a publicly-routable address the administrator of the site associates with a fully-qualified domain name, such as www.example.com, and which is assigned to one or more *virtual server*[1]. Note that a VIP address migrates from one LVS router to the other during a failover, thus maintaining a presence at that IP address, also known as *floating IP addresses*.

VIP addresses may be aliased to the same device which connects the LVS router to the Internet. For instance, if eth0 is connected to the Internet, than multiple virtual servers can be aliased to `eth0:1`. Alternatively, each virtual server can be associated with a separate device per service. For example, HTTP traffic can be handled on `eth0:1`, and FTP traffic can be handled on `eth0:2`.

Only one LVS router is active at a time. The role of the active router is to redirect service requests from virtual IP addresses to the real servers. The redirection is based on one of eight supported load-balancing algorithms described further in Section 7.3 *LVS Scheduling Overview*.

[1] A virtual server is a service configured to listen on a specific virtual IP. Refer to Section 10.6 ***VIRTUAL SERVERS*** for more on configuring a virtual server using the **Piranha Configuration Tool**.

The active router also dynamically monitors the overall health of the specific services on the real servers through simple *send/expect scripts*. To aid in detecting the health of services that require dynamic data, such as HTTPS or SSL, the administrator can also call external executables. If a service on a real server malfunctions, the active router stops sending jobs to that server until it returns to normal operation.

The backup router performs the role of a standby system. Periodically, the LVS routers exchange heartbeat messages through the primary external public interface and, in a failover situation, the private interface. Should the backup node fail to receive a heartbeat message within an expected interval, it initiates a failover and assumes the role of the active router. During failover, the backup router takes over the VIP addresses serviced by the failed router using a technique known as *ARP spoofing* — where the backup LVS router announces itself as the destination for IP packets addressed to the failed node. When the failed node returns to active service, the backup node assumes its hot-backup role again.

The simple, two-layered configuration used in Figure 7-1 is best for clusters serving data which does not change very frequently — such as static webpages — because the individual real servers do not automatically sync data between each node.

7.1.1. Data Replication and Data Sharing Between Real Servers

Since there is no built-in component in LVS clustering to share the same data between the real servers, the administrator has two basic options:

- Synchronize the data across the real server pool
- Add a third layer to the topology for shared data access

The first option is preferred for servers that do not allow large numbers of users to upload or change data on the real servers. If the cluster allows large numbers of users to modify data, such as an e-commerce website, adding a third layer is preferable.

7.1.1.1. Configuring Real Servers to Synchronize Data

There are many ways an administrator can choose to synchronize data across the pool of real servers. For instance, shell scripts can be employed so that if a Web engineer updates a page, the page is posted to all of the servers simultaneously. Also, the cluster administrator can use programs such as `rsync` to replicate changed data across all nodes at a set interval.

However, this type of data synchronization does not optimally function if the cluster is overloaded with users constantly uploading files or issuing database transactions. For a cluster with a high load, a *three-tiered topology* is the ideal solution.

7.2. A Three Tiered LVS Configuration

Figure 7-2 shows a typical three tiered LVS cluster topology. In this example, the active LVS router routes the requests from the Internet to the pool of real servers. Each of the real servers then accesses a shared data source over the network.

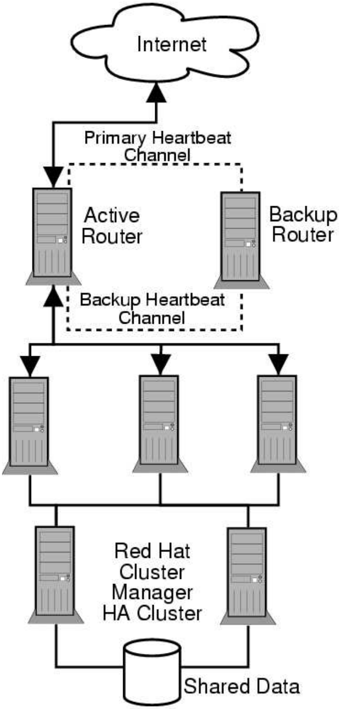


Figure 7-2. A Three Tiered LVS Configuration

This configuration is ideal for busy FTP servers, where accessible data is stored on a central, highly available server and accessed by each real server via an exported NFS directory or Samba share. This topography is also recommended for websites that access a central, highly available database for transactions. Additionally, using an active-active configuration with Red Hat Cluster Manager, administrators can configure one high-availability cluster to serve both of these roles simultaneously.

The third tier in the above example does not have to use Red Hat Cluster Manager, but failing to use a highly available solution would introduce a critical single point of failure.

7.3. LVS Scheduling Overview

One of the advantages of using an LVS cluster is its ability to perform flexible, IP-level load balancing on the real server pool. This flexibility is due to the variety of scheduling algorithms an administrator can choose from when configuring a cluster. LVS load balancing is superior to less flexible methods, such as *Round-Robin DNS* where the hierarchical nature of DNS and the caching by client machines can lead to load imbalances. Additionally, the low-level filtering employed by the LVS router has advantages over application-level request forwarding because balancing loads at the network packet level causes minimal computational overhead and allows for greater scalability.

Using scheduling, the active router can take into account the real servers' activity and, optionally, an administrator-assigned *weight* factor when routing service requests. Using assigned weights gives arbitrary priorities to individual machines. Using this form of scheduling, it is possible to create a group of real servers using a variety of hardware and software combinations and the active router can evenly load each real server.

The scheduling mechanism for an LVS cluster is provided by a collection of kernel patches called *IP Virtual Server* or *IPVS* modules. These modules enable *layer 4* (*L4*) transport layer switching, which is designed to work well with multiple servers on a single IP address.

To track and route packets to the real servers efficiently, IPVS builds an *IPVS table* in the kernel. This table is used by the active LVS router to redirect requests from a virtual server address to and returning from real servers in the pool. The IPVS table is constantly updated by a utility called *ipvsadm* — adding and removing cluster members depending on their availability.

7.3.1. Scheduling Algorithms

The structure that the IPVS table takes depends on the scheduling algorithm that the administrator chooses for any given virtual server. To allow for maximum flexibility in the types of services you can cluster and how these services are scheduled, Red Hat Enterprise Linux provides the following scheduling algorithms listed below. For instructions on how to assign scheduling algorithms refer to Section 10.6.1 *The **VIRTUAL SERVER** Subsection*.

Round-Robin Scheduling

> Distributes each request sequentially around the pool of real servers. Using this algorithm, all the real servers are treated as equals without regard to capacity or load. This scheduling model resembles round-robin DNS but is more granular due to the fact

that it is network-connection based and not host-based. LVS round-robin scheduling also does not suffer the imbalances caused by cached DNS queries.

Weighted Round-Robin Scheduling

Distributes each request sequentially around the pool of real servers but gives more jobs to servers with greater capacity. Capacity is indicated by a user-assigned weight factor, which is then adjusted upward or downward by dynamic load information. Refer to Section 7.3.2 *Server Weight and Scheduling* for more on weighting real servers.

Weighted round-robin scheduling is a preferred choice if there are significant differences in the capacity of real servers in the pool. However, if the request load varies dramatically, the more heavily weighted server may answer more than its share of requests.

Least-Connection

Distributes more requests to real servers with fewer active connections. Because it keeps track of live connections to the real servers through the IPVS table, least-connection is a type of dynamic scheduling algorithm, making it a better choice if there is a high degree of variation in the request load. It is best suited for a real server pool where each member node has roughly the same capacity. If a group of servers have different capabilities, weighted least-connection scheduling is a better choice.

Weighted Least-Connections (default)

Distributes more requests to servers with fewer active connections relative to their capacities. Capacity is indicated by a user-assigned weight, which is then adjusted upward or downward by dynamic load information. The addition of weighting makes this algorithm ideal when the real server pool contains hardware of varying capacity. Refer to Section 7.3.2 *Server Weight and Scheduling* for more on weighting real servers.

Locality-Based Least-Connection Scheduling

Distributes more requests to servers with fewer active connections relative to their destination IPs. This algorithm is designed for use in a proxy-cache server cluster. It routes the packets for an IP address to the server for that address unless that server is above its capacity and has a server in its half load, in which case it assigns the IP address to the least loaded real server.

Locality-Based Least-Connection Scheduling with Replication Scheduling

Distributes more requests to servers with fewer active connections relative to their destination IPs. This algorithm is also designed for use in a proxy-cache server cluster. It differs from Locality-Based Least-Connection Scheduling by mapping the target IP address to a subset of real server nodes. Requests are then routed to the server in this subset with the lowest number of connections. If all the nodes for the destination IP are above capacity, it replicates a new server for that destination IP address by adding the real server with the least connections from the overall pool of real servers to the subset of real servers for that destination IP. The most loaded node is then dropped from the real server subset to prevent over-replication.

Destination Hash Scheduling

Distributes requests to the pool of real servers by looking up the destination IP in a static hash table. This algorithm is designed for use in a proxy-cache server cluster.

Source Hash Scheduling

Distributes requests to the pool of real servers by looking up the source IP in a static hash table. This algorithm is designed for LVS routers with multiple firewalls.

7.3.2. Server Weight and Scheduling

The administrator of an LVS cluster can assign a *weight* to each node in the real server pool. This weight is an integer value which is factored into any *weight-aware* scheduling algorithms (such as weighted least-connections) and helps the LVS router more evenly load hardware with different capabilities.

Weights work as a ratio relative to one another. For instance, if one real server has a weight of 1 and the other server has a weight of 5, then the server with a weight of 5 gets 5 connections for every 1 connection the other server gets. The default value for a real server weight is 1.

Although adding weight to varying hardware configurations in a real server pool can help load-balance the cluster more efficiently, it can cause temporary imbalances when a real server is introduced to the real server pool and the virtual server is scheduled using weighted least-connections. For example, suppose there are three servers in the real server pool. Servers A and B are weighted at 1 and the third, server C, is weighted at 2. If server C goes down for any reason, servers A and B evenly distributes the abandoned load. However, once server C comes back online, the LVS router sees it has zero connections and floods the server with all incoming requests until it is on par with servers A and B.

To prevent this phenomenon, administrators can make the virtual server a *quiesce* server — anytime a new real server node comes online, the least-connections table is reset to zero and the LVS router routes requests as if all the real servers were newly added to the cluster.

7.4. Routing Methods

Red Hat Enterprise Linux uses *Network Address Translation* or *NAT routing* for LVS clustering, which allows the administrator tremendous flexibility when utilizing available hardware and integrating the cluster into an existing network.

7.4.1. NAT Routing

Figure 7-3, illustrates an LVS cluster utilizing NAT routing to move requests between the Internet and a private network.

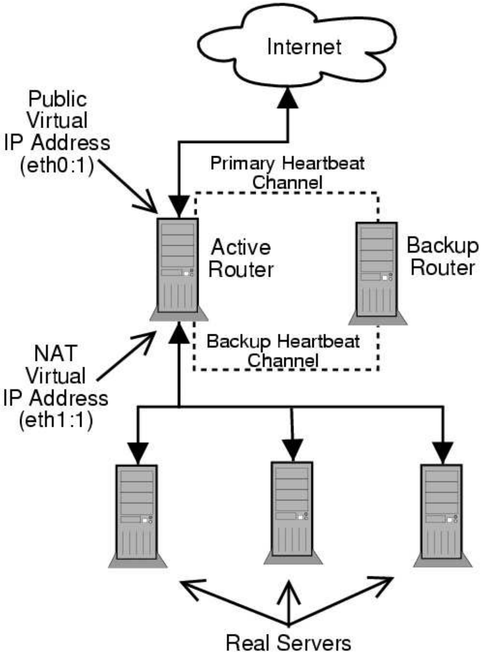

Internet
Public Virtual IP Address (eth0:1)
Primary Heartbeat Channel
Active Router
Backup Router
NAT Virtual IP Address (eth1:1)
Backup Heartbeat Channel
Real Servers

Figure 7-3. An LVS Cluster Implemented with NAT Routing

In the example, there are two NICs in the active LVS router. The NIC for the Internet has a *real IP address* on eth0 and has a floating IP address aliased to eth0:1. The NIC for the private network interface has a real IP address on eth1 and has a floating IP address aliased to eth1:1. In the event of failover, the virtual interface facing the Internet and the private facing virtual interface are taken-over by the backup LVS router simultaneously. All of the cluster's real servers located on the private network use the floating IP for the NAT router as their default route to communicate with the active LVS router so that their abilities to respond to requests from the Internet is not impaired.

In this example, the LVS router's public LVS floating IP address and private NAT floating IP address are aliased to two physical NICs. While it is possible to associate each floating IP address to its own physical device on the LVS router nodes, having more than two NICs is not a requirement.

Using this topography, the active LVS router receives the request and routes it to the appropriate server. The real server then processes the request and returns the packets to the LVS router which uses network address translation to replace the address of the real server in the packets with the LVS routers public VIP address. This process is called *IP masquerading* because the actual IP addresses of the real servers is hidden from the requesting clients.

Using this NAT routing, the real servers may be any kind of machine running various operating systems. The main disadvantage is that the LVS router may become a bottleneck in large cluster deployments because it must process outgoing as well as incoming requests.

7.5. Persistence and Firewall Marks

In certain situations, it may be desirable for a client to reconnect repeatedly to the same real server, rather than have an LVS load balancing algorithm send that request to the best available server. Examples of such situations include multi-screen web forms, cookies, SSL, and FTP connections. In these cases, a client may not work properly unless the transactions are being handled by the same server to retain context. LVS provides two different features to handle this: *persistence* and *firewall marks*.

7.5.1. Persistence

When enabled, persistence acts like a timer. When a client connects to a service, LVS remembers the last connection for a specified period of time. If that same client IP address connects again within that period, it is sent to the same server it connected to previously — bypassing the load-balancing mechanisms. When a connection occurs outside the time window, it is handled according to the scheduling rules in place.

Persistence also allows the administrator to specify a subnet mask to apply to the client IP address test as a tool for controlling what addresses have a higher level of persistence, thereby grouping connections to that subnet.

Grouping connections destined for different ports can be important for protocols which use more than one port to communicate, such as FTP. However, persistence is not the most efficient way to deal with the problem of grouping together connections destined for different ports. For these situations, it is best to use *firewall marks*.

7.5.2. Firewall Marks

Firewall marks are an easy and efficient way to a group ports used for a protocol or group of related protocols. For instance, if an LVS cluster is deployed to run an e-commerce site, firewall marks can be used to bundle HTTP connections on port 80 and secure, HTTPS connections on port 443. By assigning the same firewall mark to the virtual server for each protocol, state information for the transaction can be preserved because the LVS router forwards all requests to the same real server after a connection is opened.

Because of its efficiency and ease-of-use, administrators of LVS clusters should use firewall marks instead of persistence whenever possible for grouping connections. However, administrators should still add persistence to the virtual servers in conjunction with firewall marks to ensure the clients are reconnected to the same server for an adequate period of time.

7.6. LVS Cluster — A Block Diagram

LVS routers use a collection of programs to monitor cluster members and cluster services. Figure 7-4 illustrates how these various programs on both the active and backup LVS routers work together to manage the cluster.

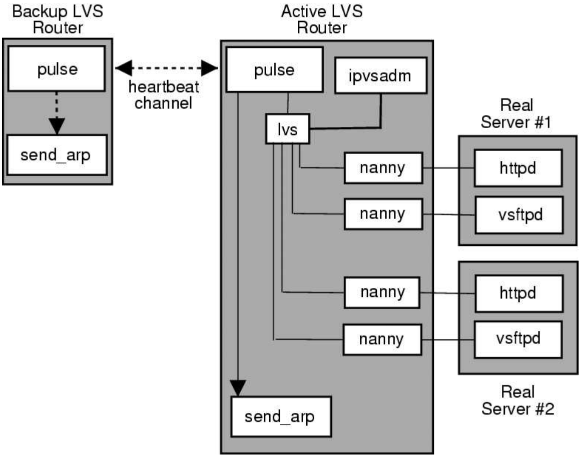


Figure 7-4. Components of a Running LVS Cluster

The `pulse` daemon runs on both the active and passive LVS routers. On the backup router, `pulse` sends a *heartbeat* to the public interface of the active router to make sure the active router is still properly functioning. On the active router, `pulse` starts the `lvs` daemon and responds to *heartbeat* queries from the backup LVS router.

Once started, the `lvs` daemon calls the `ipvsadm` utility to configure and maintain the IPVS routing table in the kernel and starts a `nanny` process for each configured virtual server on each real server. Each `nanny` process checks the state of one configured service on one real server, and tells the `lvs` daemon if the service on that real server is malfunctioning. If a malfunction is detected, the `lvs` daemon instructs `ipvsadm` to remove that real server from the IPVS routing table.

If the backup router does not receive a response from the active router, it initiates failover by calling `send_arp` to reassign all virtual IP addresses to the NIC hardware addresses (*MAC* address) of the backup node, sends a command to the active router via both the public and private network interfaces to shut down the `lvs` daemon on the active router, and starts the `lvs` daemon on the backup node to accept requests for the configured virtual servers.

7.6.1. Components of an LVS Cluster

Section 7.6.1.1 *`pulse`* shows a detailed list of each software component in an LVS router.

7.6.1.1. `pulse`

This is the controlling process which starts all other daemons related to LVS routers. At boot time, the daemon is started by the `/etc/rc.d/init.d/pulse` script. It then reads the configuration file `/etc/sysconfig/ha/lvs.cf`. On the active router, `pulse` starts the LVS daemon. On the backup router, `pulse` determines the health of the active router by executing a simple heartbeat at a user-configurable interval. If the active router fails to respond after a user-configurable interval, it initiates failover. During failover, `pulse` on the backup router instructs the `pulse` daemon on the active router to shut down all LVS services, starts the `send_arp` program to reassign the floating IP addresses to the backup router’s MAC address, and starts the `lvs` daemon.

7.6.1.2. `lvs`

The `lvs` daemon runs on the active LVS router once called by `pulse`. It reads the configuration file `/etc/sysconfig/ha/lvs.cf`, calls the `ipvsadm` utility to build and maintain the IPVS routing table, and assigns a `nanny` process for each configured LVS service. If `nanny` reports a real server is down, `lvs` instructs the `ipvsadm` utility to remove the real server from the IPVS routing table.

7.6.1.3. ipvsadm

This service updates the IPVS routing table in the kernel. The lvs daemon sets up and administers an LVS cluster by calling ipvsadm to add, change, or delete entries in the IPVS routing table.

7.6.1.4. nanny

The nanny monitoring daemon runs on the active LVS router. Through this daemon, the active router determines the health of each real server and, optionally, monitors its workload. A separate process runs for each service defined on each real server.

7.6.1.5. /etc/sysconfig/ha/lvs.cf

This is the LVS cluster configuration file. Directly or indirectly, all daemons get their configuration information from this file.

7.6.1.6. Piranha Configuration Tool

This is the Web-based tool for monitoring, configuring, and administering an LVS cluster. This is the default tool to maintain the /etc/sysconfig/ha/lvs.cf LVS cluster configuration file.

7.6.1.7. send_arp

This program sends out ARP broadcasts when the floating IP address changes from one node to another during failover.

Chapter 8 *Initial LVS Configuration* reviews important post-installation configuration steps you should take before configuring Red Hat Enterprise Linux to be an LVS router.

Chapter 8.

Initial LVS Configuration

After installing Red Hat Enterprise Linux, you must take some basic steps to set up both the LVS routers and the real servers in the LVS cluster. This chapter covers these initial steps in detail.

Note

The LVS router node that becomes the active node once the cluster is started is also referred to as the *primary node*. When configuring an LVS cluster, use the **Piranha Configuration Tool** on the primary node.

8.1. Configuring Services on the LVS Routers

The Red Hat Enterprise Linux installation program installs all of the components needed to set up an LVS cluster, but the appropriate services must be activated before configuring the cluster. For both LVS routers, set the appropriate services to start at boot time. There are three primary tools available for setting services to activate at boot time under Red Hat Enterprise Linux: the command line program `chkconfig`, the ncurses-based program `ntsysv`, and the graphical **Services Configuration Tool**. All of these tools require root access.

Tip

To attain root access, open a shell prompt and type the following command followed by the root password:

```
su -
```

On the LVS routers, there are three services which need to be set to activate at boot time:

- The `piranha-gui` service (primary node only)
- The `pulse` service
- The `sshd` service

If you are clustering multi-port services or using firewall marks, you must also enable the `iptables` service.

It is best to set these services to activate in both runlevel 3 and runlevel 5. To accomplish this using `chkconfig`, type the following command for each service:

```
/sbin/chkconfig --level 35 daemon on
```

In the above command, replace *daemon* with the name of the service you are activating. To get a list of services on the system as well as what runlevel they are set to activate on, issue the following command:

```
/sbin/chkconfig --list
```

Warning

> Turning any of the above services on using `chkconfig` does not actually start the daemon. To do this use the `/sbin/service` command. See Section 8.3 *Starting the **Piranha Configuration Tool** Service* for an example of how to use the `/sbin/service` command.

For more information on runlevels and configuring services with `ntsysv` and the **Services Configuration Tool**, refer to the chapter titled *"Controlling Access to Services"* in the *Red Hat Enterprise Linux System Administration Guide*.

8.2. Setting a Password for the Piranha Configuration Tool

Before using the **Piranha Configuration Tool** for the first time on the primary LVS router, you must restrict access to it by creating a password. To do this, login as root and issue the following command:

```
/usr/sbin/piranha-passwd
```

After entering this command, create the administrative password when prompted.

Warning

> For a password to be more secure, it should not contain proper nouns, commonly used acronyms, or words in a dictionary from any language. Do not leave the password unencrypted anywhere on the system.

If the password is changed during an active **Piranha Configuration Tool** session, the administrator is prompted to provide the new password.

8.3. Starting the Piranha Configuration Tool Service

After you have set the password for the **Piranha Configuration Tool**, start or restart the `piranha-gui` service located in `/etc/rc.d/init.d/piranha-gui`. To do this, type the following command as root:

```
/sbin/service piranha-gui start
```

or

```
/sbin/service piranha-gui restart
```

Issuing this command starts a private session of the Apache HTTP Server by calling the symbolic link `/usr/sbin/piranha_gui -> /usr/sbin/httpd`. For security reasons, the `piranha-gui` version of `httpd` runs as the piranha user in a separate process. The fact that `piranha-gui` leverages the `httpd` service means that:

1. The Apache HTTP Server must be installed on the system.
2. Stopping or restarting the Apache HTTP Server via the `service` command stops the `piranha-gui` service.

Warning

> If the command `/sbin/service httpd stop` or `/sbin/service httpd restart` is issued on an LVS router, you must start the `piranha-gui` service by issuing the following command:
>
> ```
> /sbin/service piranha-gui start
> ```

The `piranha-gui` service is all that is necessary to begin configuring an LVS cluster. However, if you are configuring the cluster remotely, the `sshd` service is also required. You do *not* need to start the `pulse` service until configuration using the **Piranha Configuration Tool** is complete. See Section 10.8 *Starting the Cluster* for information on starting the `pulse` service.

8.3.1. Configuring the Piranha Configuration Tool Web Server Port

The **Piranha Configuration Tool** runs on port 3636 by default. To change this port number, change the line `Listen 3636` in Section 2 of the `piranha-gui` Web server configuration file `/etc/sysconfig/ha/conf/httpd.conf`.

To use the **Piranha Configuration Tool** you need at minimum a text-only Web browser. If you start a Web browser on the primary LVS router, open the location **`http://`*`localhost`*`:3636`**. You can reach the **Piranha Configuration Tool** from anywhere via Web browser by replacing *`localhost`* with the hostname or IP address of the primary LVS router.

When your browser connects to the **Piranha Configuration Tool**, you must login to access the cluster configuration services. Enter **`piranha`** in the **Username** field and the password set with `piranha-passwd` in the **Password** field.

Now that the **Piranha Configuration Tool** is running, you may wish to consider limiting who has access to the tool over the network. The next section reviews ways to accomplish this task.

8.4. Limiting Access To the Piranha Configuration Tool

The **Piranha Configuration Tool** prompts for a valid username and password combination. However, because all of the data passed to the **Piranha Configuration Tool** is in plain text, it is recommended that you restrict access only to trusted networks or to the local machine.

The easiest way to restrict access is to use the Apache HTTP Server's built in access control mechanisms by editing `/etc/sysconfig/ha/web/secure/.htaccess`. After altering the file you do not have to restart the `piranha-gui` service because the server checks the `.htaccess` file each time it accesses the directory.

By default, the access controls for this directory allow anyone to view the contents of the directory. Here is what the default access looks like:

```
Order deny,allow
Allow from all
```

To limit access of the **Piranha Configuration Tool** to only the localhost change the `.htaccess` file to allow access from only the loopback device (127.0.0.1). For more information on the loopback device, see the chapter titled *Network Scripts* in the *Red Hat Enterprise Linux Reference Guide*.

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

You can also allow specific hosts or subnets as seen in this example:

```
Order deny,allow
Deny from all
Allow from 192.168.1.100
Allow from 172.16.57
```

In this example, only Web browsers from the machine with the IP address of 192.168.1.100 and machines on the 172.16.57/24 network can access the **Piranha Configuration Tool**.

Caution

> Editing the **Piranha Configuration Tool** `.htaccess` file limits access to the configuration pages in the `/etc/sysconfig/ha/web/secure/` directory but not to the login and the help pages in `/etc/sysconfig/ha/web/`. To limit access to this directory, create a `.htaccess` file in the `/etc/sysconfig/ha/web/` directory with **`order`**, **`allow`**, and **`deny`** lines identical to `/etc/sysconfig/ha/web/secure/.htaccess`.

8.5. Turning on Packet Forwarding

In order for the LVS router to forward network packets properly to the real servers, each LVS router node must have IP forwarding turned on in the kernel. Log in as root and change the line which reads `net.ipv4.ip_forward = 0` in `/etc/sysctl.conf` to the following:

```
net.ipv4.ip_forward = 1
```

The changes take effect when you reboot the system.

To check if IP forwarding is turned on, issue the following command as root:

```
/sbin/sysctl net.ipv4.ip_forward
```

If the above command returns a `1`, then IP forwarding is enabled. If it returns a `0`, then you can turn it on manually using the following command:

```
/sbin/sysctl -w net.ipv4.ip_forward=1
```

8.6. Configuring Services on the Real Servers

If the real servers in the cluster are Red Hat Enterprise Linux systems, set the appropriate server daemons to activate at boot time. These daemons can include `httpd` for Web services or `xinetd` for FTP or Telnet services.

It may also be useful to access the real servers remotely, so the `sshd` daemon should also be installed and running.

Chapter 9.

Setting Up a Red Hat Enterprise Linux LVS Cluster

A Red Hat Enterprise Linux LVS cluster consists of two basic groups: the LVS routers and the real servers. To prevent a single point of failure, each groups should contain at least two member systems.

The LVS router group should consist of two identical or very similar systems running Red Hat Enterprise Linux. One will act as the active LVS router while the other stays in hot standby mode, so they need to have as close to the same capabilities as possible.

Before choosing and configuring the hardware for the real server group, you most decide what which of the three types of LVS topographies to use.

9.1. The NAT LVS Cluster

The NAT topography allows for great latitude in utilizing existing hardware, but it is limited in its ability to handle large loads due to the fact that all packets going into and coming out of the cluster pass through the LVS router.

Network Layout

> The topography for an LVS cluster utilizing NAT routing is the easiest to configure from a network layout perspective because the cluster needs only one access point to the public network. The real servers pass all requests back through the LVS router so they are on their own private network.

Hardware

> The NAT topography is the most flexible in regards to cluster hardware because the real servers do not need to be Linux machines to function correctly in the cluster. In a NAT cluster, each real server only needs one NIC since it will only be responding to the LVS router. The LVS routers, on the other hand, need two NICs each to route traffic between the two networks. Because this topography creates a network bottleneck at the LVS router, gigabit Ethernet NICs can be employed on each LVS router to increase the bandwidth the LVS routers can handle. If gigabit Ethernet is employed on the LVS routers, any switch connecting the real servers to the LVS routers must have at least two gigabit Ethernet ports to handle the load efficiently.

Software

> Because the NAT topography requires the use of `iptables` for some configurations, there can be a fair amount of software configuration outside of **Piranha Configu-**

ration Tool. In particular, FTP services and the use of firewall marks requires extra manual configuration of the LVS routers to route requests properly.

9.1.1. Configuring Network Interfaces for a NAT LVS Cluster

To set up a NAT LVS cluster, the administrator must first configure the network interfaces for the public network and the private network on the LVS routers. In this example, the LVS routers' public interfaces (`eth0`) will be on the 192.168.26/24 network (I know, I know, this is not a routable IP, but let us pretend there is a firewall in front of the LVS router for good measure) and the private interfaces which link to the real servers (`eth1`) will be on the 10.11.12/24 network.

So on the active or *primary* LVS router node, the public interface's network script, `/etc/sysconfig/network-scripts/ifcfg-eth0`, could look something like this:

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.26.9
NETMASK=255.255.255.0
GATEWAY=192.168.26.254
```

The `/etc/sysconfig/network-scripts/ifcfg-eth1` for the private NAT interface on the LVS router could look something like this:

```
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.11.12.9
NETMASK=255.255.255.0
```

In this example, the VIP for the LVS router's public interface will be 192.168.26.10 and the VIP for the NAT or private interface will be 10.11.12.10. So, it is essential that the real servers route requests back to the VIP for the NAT interface.

Important

The sample Ethernet interface configuration settings in this section are for the real IP addresses of an LVS router and *not* the floating IP addresses. To configure the public and private floating IP addresses the administrator should use the **Piranha Configuration Tool**, as shown in Section 10.4 ***GLOBAL SETTINGS*** and Section 10.6.1 *The **VIRTUAL SERVER** Subsection*.

After configuring the primary LVS router node's network interfaces, configure the backup LVS router's real network interfaces — taking care that none of the IP address conflict with any other IP addresses on the network.

Important

Be sure each interface on the backup node services the same network as the interface on primary node. For instance, if eth0 connects to the public network on the primary node, it must also connect to the public network on the backup node as well.

9.1.2. Routing on the Real Servers

The most important thing to remember when configuring the real servers network interfaces in a NAT cluster is to set the gateway for the NAT floating IP address of the LVS router. In this example, that address will be 10.11.12.10.

Note

Once the network interfaces are up on the real servers, the machines will be unable to ping or connect in other ways to the public network. This is normal. You will, however, be able to ping the real IP for the LVS router's private interface, in this case 10.11.12.8.

So the real server's `/etc/sysconfig/network-scripts/ifcfg-eth0` file could look similar to this:

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.11.12.1
NETMASK=255.255.255.0
GATEWAY=10.11.12.10
```

Warning

If a real server has more than one network interface configured with a `GATEWAY=` line, the first one to come up will get the gateway. Therefore if both `eth0` and `eth1` are configured and `eth1` is used for LVS clustering, the real servers may not route requests properly.

It is best to turn off extraneous network interfaces by setting `ONBOOT=`**`no`** in their network scripts within the `/etc/sysconfig/network-scripts/` directory or by making sure the gateway is correctly set in the interface which comes up first.

9.1.3. Enabling NAT Routing on the LVS Routers

In a simple NAT LVS cluster where each clustered service uses only one port, like HTTP on port 80, the administrator needs only to enable packet forwarding on the LVS routers for the requests to be properly routed between the outside world and the real servers. See Section 8.5 *Turning on Packet Forwarding* for instructions on turning on packet forwarding. However, more configuration is necessary when the clustered services require more than one port to go to the same real server during a user session. For information on creating multi-port services using firewall marks, see Section 9.3 *Multi-port Services and LVS Clustering*.

Once forwarding is enabled on the LVS routers and the real servers are set up and have the clustered services running, use the **Piranha Configuration Tool** to configure the cluster as shown in Chapter 10 *Configuring the LVS Routers with **Piranha Configuration Tool***.

Warning

Do not configure the floating IP for `eth0:1` or `eth1:1` by manually editing network scripts or using a network configuration tool. Instead, use the **Piranha Configuration Tool** as shown in Section 10.4 ***GLOBAL SETTINGS*** and Section 10.6.1 *The **VIRTUAL SERVER** Subsection* to configure any cluster-related virtual interfaces.

When finished, start the `pulse` service as shown in Section 10.8 *Starting the Cluster*. Once `pulse` is up and running, the active LVS router will begin routing requests to the pool of real servers.

9.2. Putting the Cluster Together

After determining which of the above routing methods to use, the hardware for the LVS cluster should be linked together on the network.

Important

The adapter devices on the LVS routers must be configured to access the same networks. For instance if `eth0` connects to public network and `eth1` connects to the private network, then these same devices on the backup LVS router must connect to the same networks.

Also the gateway listed in the first interface to come up at boot time is added to the routing table and subsequent gateways listed in other interfaces are ignored. This is especially important to consider when configuring the real servers.

After physically connecting together the cluster hardware, configure the network interfaces on the primary and backup LVS routers. This can be done using a graphical application such as **system-config-network** or by editing the network scripts manually. For more information about adding devices using **system-config-network**, see the chapter titled *Network Configuration* in the *Red Hat Enterprise Linux System Administration Guide*. For more information on editing network scripts by hand, see the chapter titled *Network Scripts* in the *Red Hat Enterprise Linux Reference Guide*. For the remainder of the chapter, example alterations to network interfaces are made either manually or through the **Piranha Configuration Tool**.

9.2.1. General LVS Networking Tips

Configure the real IP addresses for both the public and private networks on the LVS routers before attempting to configure the cluster using the **Piranha Configuration Tool**. The sections on each topography give example network addresses, but the actual network addresses are needed. Below are some useful commands for bringing up network interfaces or checking their status.

Bringing Up Real Network Interfaces

The best way to bring up any real network interface is to use the following commands as root replacing *N* with the number corresponding to the interface (`eth0` and `eth1`):

```
/sbin/ifup ethN
```

Warning

Do *not* use the `ifup` scripts to bring up any floating IP addresses you may configure using **Piranha Configuration Tool** (`eth0:1` or `eth1:1`). Use the `service` command to start `pulse` instead (see Section 10.8 *Starting the Cluster* for details).

To bring a network interface down, type the following command:

```
/sbin/ifdown ethN
```

Again, replace *N* in the above command with the number corresponding to the interface you wish to bring down.

Checking the Status of Network Interfaces

If you need to check which network interfaces are up at any given time, type the following:

```
/sbin/ifconfig
```

To view the routing table for a machine, issue the following command:

```
/sbin/route
```

9.3. Multi-port Services and LVS Clustering

LVS routers under any topology require extra configuration when creating multi-port LVS services. Multi-port services can be created artificially by using firewall marks to bundle together different, but related protocols, such as HTTP (port 80) and HTTPS (port 443), or when LVS is used to cluster true multi-port protocols, such as FTP. In either case, the LVS router uses firewall marks to recognize that packets destined for different ports, but bearing the same firewall mark, should be handled identically. Also, when combined with persistence, firewall marks ensure connections from the client machine are routed to the same host, as long as the connections occur within the length of time specified by the persistence parameter. For more on assigning persistence to a virtual server, see Section 10.6.1 *The **VIRTUAL SERVER** Subsection*.

Unfortunately, the mechanism used to balance the loads on the real servers — IPVS — can recognize the firewall marks assigned to a packet, but cannot itself assign firewall marks. The job of *assigning* firewall marks must be performed by the network packet filter, `iptables`, outside of **Piranha Configuration Tool**.

9.3.1. Assigning Firewall Marks

To assign firewall marks to a packet destined for a particular port, the administrator must use `iptables`.

This section illustrates how to bundle HTTP and HTTPS as an example, however FTP is another commonly clustered multi-port protocol. If an LVS cluster is used for FTP services, see Section 9.4 *FTP In an LVS Cluster* for details on how to best configure the cluster.

The basic rule to remember when using firewall marks is that for every protocol using a firewall mark in **Piranha Configuration Tool** there must be a commensurate `iptables` rule to assign marks to the network packets.

Before creating network packet filter rules, make sure there are no rules already in place. To do this, open a shell prompt, login as root, and type:

```
/sbin/service iptables status
```

If `iptables` is not running, the prompt will instantly reappear.

If `iptables` is active, it displays a set of rules. If rules are present, type the following command:

```
/sbin/service iptables stop
```

If the rules already in place are important, check the contents of `/etc/sysconfig/iptables` and copy any rules worth keeping to a safe place before proceeding.

Below are rules which assign the same firewall mark, 80, to incoming traffic destined for the floating IP address, *n.n.n.n*, on ports 80 and 443. For instructions on assigning the VIP to the public network interface, see Section 10.6.1 *The **VIRTUAL SERVER** Subsection*. Also note that you must log in as root and load the module for `iptables` before issuing rules for the first time.

```
/sbin/modprobe ip_tables
/sbin/iptables -t mangle -A PREROUTING -p tcp \
     -d n.n.n.n/32 --dport 80 -j MARK --set-mark 80
/sbin/iptables -t mangle-A PREROUTING -p tcp \
              -d n.n.n.n/32 --dport 443 -j MARK --set-mark 80
```

In the above `iptables` commands, *n.n.n.n* should be replaced with the floating IP for your HTTP and HTTPS virtual servers. These commands have the net effect of assigning any traffic addressed to the VIP on the appropriate ports a firewall mark of 80, which in turn is recognized by IPVS and forwarded appropriately.

⚠ **Warning**

> The commands above will take effect immediately, but do not persist through a reboot of the system. To ensure network packet filter settings are restored upon reboot, refer to Section 9.5 *Saving Network Packet Filter Settings*

9.4. FTP In an LVS Cluster

File Transport Protocol (FTP) is an old and complex multi-port protocol that presents a distinct set of challenges to a clustered environment. To understand the nature of these challenges, you must first understand some key things about how FTP works.

9.4.1. How FTP Works

With most other server client relationships, the client machine opens up a connection to the server on a particular port and the server then responds to the client on that port. When an FTP client connects to an FTP server it opens a connection to the FTP control port 21. Then the *client* tells the FTP *server* whether to establish an *active* or *passive* connection. The type of connection chosen by the client determines how the server responds and on what ports transactions will occur.

The two types of data connections are:

Active Connections

> When an active connection is established, the *server* opens a data connection to the client from port 20 to a high range port on the client machine. All data from the server is then passed over this connection.

Passive Connections

> When a passive connection is established, the *client* asks the FTP server to establish a passive connection port, which can be on any port higher than 10,000. The server then binds to this high-numbered port for this particular session and relays that port number back to the client. The client then opens the newly bound port for the data connection. Each data request the client makes results in a separate data connection. Most modern FTP clients attempt to establish a passive connection when requesting data from servers.

The two important things to note about all of this in regards to clustering is:

1. The *client* determines the type of connection, not the server. This means, to effectively cluster FTP, you must configure the LVS routers to handle both active and passive connections.
2. The FTP client/server relationship can potentially open a large number of ports that the **Piranha Configuration Tool** and IPVS do not know about.

9.4.2. How This Affects LVS Routing

IPVS packet forwarding only allows connections in and out of the cluster based on it recognizing its port number or its firewall mark. If a client from outside the cluster attempts to open a port IPVS is not configured to handle, it drops the connection. Similarly, if the real server attempts to open a connection back out to the Internet on a port IPVS does not know about, it drops the connection. This means *all* connections from FTP clients on the Internet *must* have the same firewall mark assigned to them and all connections from the FTP server *must* be properly forwarded to the Internet using network packet filtering rules.

9.4.3. Creating Network Packet Filter Rules

Before assigning any `iptables` rules for FTP service, review the information in Section 9.3.1 *Assigning Firewall Marks* concerning multi-port services and techniques for checking the existing network packet filtering rules.

Below are rules which assign the same firewall mark, 21, to FTP traffic. For these rules to work properly, you must also use the **VIRTUAL SERVER** subsection of **Piranha Configuration Tool** to configure a virtual server for port 21 with a value of **`21`** in the **Firewall Mark** field. See Section 10.6.1 *The **VIRTUAL SERVER** Subsection* for details.

9.4.3.1. Rules for Active Connections

The rules for active connections tell the kernel to accept and forward connections coming to the *internal* floating IP address on port 20 — the FTP data port.

```
iptables

    /sbin/iptables -t nat -A POSTROUTING -p tcp \
              -s n.n.n.0/24 --sport 20 -j MASQUERADE
```

In the above `iptables` commands, *`n.n.n`* should be replaced with the first three values for the floating IP for the NAT interface's internal network interface defined in the **GLOBAL SETTINGS** panel of **Piranha Configuration Tool**. The command allows the LVS router to accept outgoing connections from the real servers that IPVS does not know about.

9.4.3.2. Rules for Passive Connections

The rules for passive connections assign the appropriate firewall mark to connections coming in from the Internet to the floating IP for the service on a wide range of ports — 10,000 to 20,000.

Warning

> If you are limiting the port range for passive connections, you must also configure the VSFTP server to use a matching port range. This can be accomplished by adding the following lines to `/etc/vsftpd.conf`:
>
> ```
> pasv_min_port=10000
> pasv_max_port=20000
> ```
>
> You must also control the address that the server displays to the client for passive FTP connections. In a NAT routed LVS system, add the following line to `/etc/vsftpd.conf` to override the real server IP address to the VIP, which is what the client sees upon connection. For example:

pasv_address=*X.X.X.X*

Replace *X.X.X.X* with the VIP address of the LVS system.

For configuration of other FTP servers, consult the respective documentation.

This range should be a wide enough for most situations; however, you can increase this number to include all available non-secured ports by changing `10000:20000` in the commands below to `1024:65535`.

iptables

```
    /sbin/iptables -t mangle -A PREROUTING -p tcp \
               -d n.n.n.n/32  \
               --dport 21 -j MARK --set-mark 21
/sbin/iptables -t mangle -A PREROUTING -p tcp \
               -d n.n.n.n/32 \
               --dport 10000:20000 -j MARK --set-mark 21
```

In the above `iptables` commands, *n.n.n.n* should be replaced with the floating IP for the FTP virtual server defined in the **VIRTUAL SERVER** subsection of **Piranha Configuration Tool**. These commands have the net effect of assigning any traffic addressed to the floating IP on the appropriate ports a firewall mark of 21, which is in turn recognized by IPVS and forwarded appropriately.

Warning

The commands above take effect immediately, but do not persist through a reboot of the system. To ensure network packet filter settings are restored after a reboot, see Section 9.5 *Saving Network Packet Filter Settings*

Finally, you need to be sure that the appropriate service is set to activate on the proper runlevels. For more on this, refer to Section 8.1 *Configuring Services on the LVS Routers*.

9.5. Saving Network Packet Filter Settings

After configuring the appropriate network packet filters for your situation, save the settings so they get restored after a reboot. For `iptables`, type the following command:

```
/sbin/service iptables save
```

This saves the settings in `/etc/sysconfig/iptables` so they can be recalled at boot time.

Once this file is written, you are able to use the `/sbin/service` command to start, stop, and check the status (using the status switch) of `iptables`. The `/sbin/service` will automatically load the appropriate module for you. For an example of how to use the `/sbin/service` command, see Section 8.3 *Starting the **Piranha Configuration Tool** Service*.

Finally, you need to be sure the appropriate service is set to activate on the proper runlevels. For more on this, see Section 8.1 *Configuring Services on the LVS Routers*.

The next chapter explains how to use the **Piranha Configuration Tool** to configure the LVS router and describe the steps necessary to active an LVS cluster.

Chapter 10.

Configuring the LVS Routers with Piranha Configuration Tool

The **Piranha Configuration Tool** provides a structured approach to creating the necessary configuration file for a Piranha cluster — `/etc/sysconfig/ha/lvs.cf`. This chapter describes the basic operation of the **Piranha Configuration Tool** and how to activate the cluster once configuration is complete.

Important

> The configuration file for the LVS cluster follows strict formatting rules. Using the **Piranha Configuration Tool** is the best way to prevent syntax errors in the `lvs.cf` and therefore prevent software failures.

10.1. Necessary Software

The `piranha-gui` service must be running on the primary LVS router to use the **Piranha Configuration Tool**. To configure the cluster, you minimally need a text-only Web browser, such as `links`. If you are accessing the LVS router from another machine, you also need an `ssh` connection to the primary LVS router as the root user.

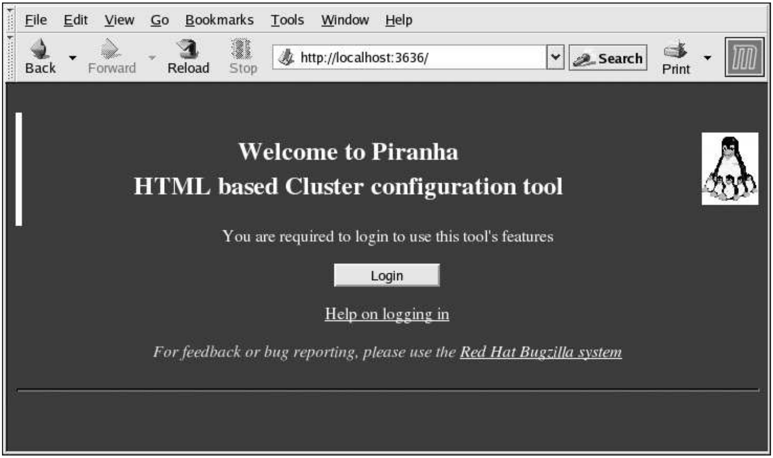
While configuring the primary LVS router it is a good idea to keep a concurrent `ssh` connection in a terminal window. This connection provides a secure way to restart `pulse` and other services, configure network packet filters, and monitor `/var/log/messages` during trouble shooting.

The next four sections walk through each of the configuration pages of the **Piranha Configuration Tool** and give instructions on using it to set up the LVS cluster.

10.2. Logging Into the Piranha Configuration Tool

When configuring an LVS cluster, you should always begin by configuring the primary router with the **Piranha Configuration Tool**. To do this,verify that the `piranha-gui` service is running and an administrative password has been set, as described in Section 8.2 *Setting a Password for the **Piranha Configuration Tool***.

If you are accessing the machine locally, you can open **`http://localhost:3636`** in a Web browser to access the **Piranha Configuration Tool**. Otherwise, type in the hostname or real IP address for the server followed by **`:3636`**. Once the browser connects, you will see the screen shown in Figure 10-1.



Welcome to Piranha
HTML based Cluster configuration tool

You are required to login to use this tool's features

For feedback or bug reporting, please use the Red Hat Bugzilla system

Figure 10-1. The Welcome Panel

Click on the **Login** button and enter **`piranha`** for the **Username** and the administrative password you created in the **Password** field.

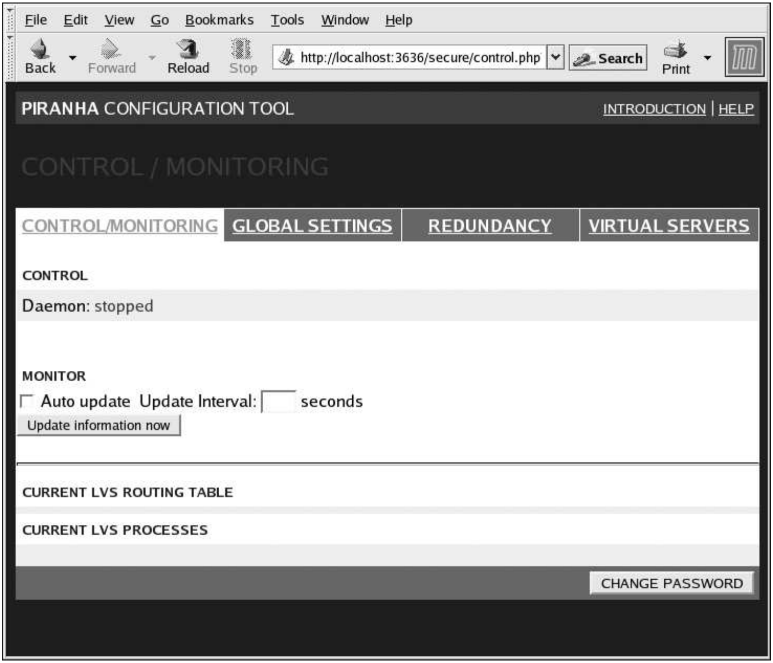
The **Piranha Configuration Tool** is made of four main screens or *panels*. In addition, the **Virtual Servers** panel contains four *subsections*. The **CONTROL/MONITORING** panel is the first panel after the login screen.

10.3. CONTROL/MONITORING

The **CONTROL/MONITORING** Panel presents the cluster administrator with a limited runtime status of the cluster. It displays the status of the `pulse` daemon, the LVS routing table, and the LVS-spawned `nanny` processes.

Note

The fields for **CURRENT LVS ROUTING TABLE** and **CURRENT LVS PROCESSES** remain blank until you actually start the cluster, as shown in Section 10.8 *Starting the Cluster*.



http://localhost:3636/secure/control.php

CONTROL

Daemon: stopped

MONITOR

□ Auto update Update Interval: ____ seconds

CURRENT LVS ROUTING TABLE

CURRENT LVS PROCESSES

Figure 10-2. The CONTROL/MONITORING Panel

Auto update

The status display on this page can be updated automatically at a user configurable interval. To enable this feature, click on the **Auto update** checkbox and set the desired update frequency in the **Update frequency in seconds** text box (the default value is 10 seconds).

It is not recommended that you set the automatic update to an interval less than 10 seconds. Doing so may make it difficult to reconfigure the **Auto update** interval because the page will update too frequently. If you encounter this issue, simply click on another panel and then back on **CONTROL/MONITORING**.

The **Auto update** feature does not work with all browsers, such as **Mozilla**.

Update information now

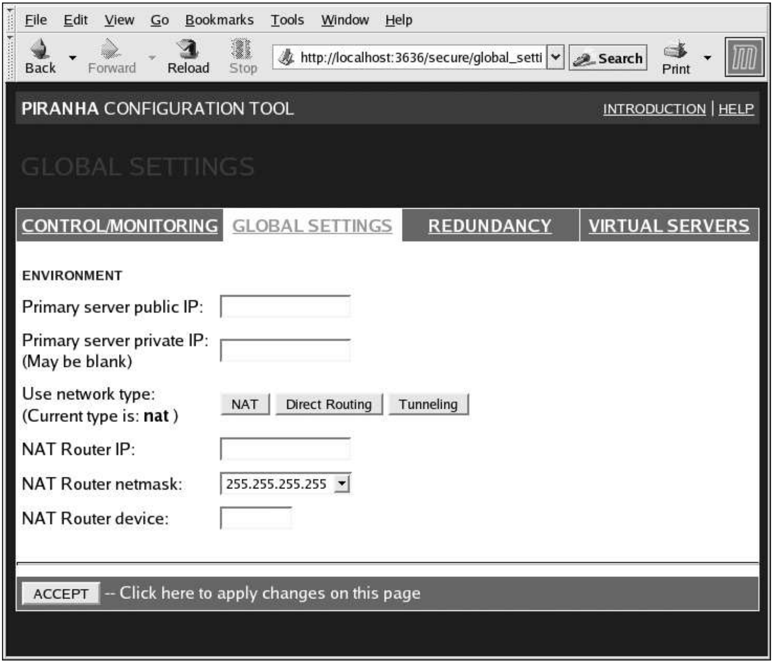
You can manually update the status information manually by clicking this button.

CHANGE PASSWORD

Clicking this button takes you to a help screen with information on how to change the administrative password for the **Piranha Configuration Tool**.

10.4. GLOBAL SETTINGS

The **GLOBAL SETTINGS** panel is where the cluster administrator defines the networking details for the primary LVS router's public and private network interfaces.



ENVIRONMENT

Primary server public IP:

Primary server private IP: (May be blank)

Use network type: (Current type is: **nat**)

NAT Router IP:

NAT Router netmask: 255.255.255.255

NAT Router device:

Figure 10-3. The GLOBAL SETTINGS Panel

The top half of this panel sets up the primary LVS router's public and private network interfaces. These are the interfaces already configured in Section 9.1.1 *Configuring Network Interfaces for a NAT LVS Cluster*.

Primary server public IP

In this field, enter the publicly routable real IP address for the primary LVS node.

Primary server private IP

Enter the real IP address for an alternative network interface on the primary LVS node. This address is used solely as an alternative heartbeat channel for the backup router and does not have to correlate to the real private IP address assigned in Section 9.1.1 *Configuring Network Interfaces for a NAT LVS Cluster*. You may leave this field blank, but doing so will mean there is no alternate heartbeat channel for the backup LVS router to use and therefore will create a single point of failure.

Tip

The primary LVS router's private IP can be configured on any interface that accepts TCP/IP, whether it be an Ethernet adapter or a serial port.

Use network type

Click the **NAT** button to select NAT routing.

The next three fields deal specifically with the NAT router's virtual network interface connected the private network with the real servers.

NAT Router IP

Enter the private floating IP in this text field. This floating IP should be used as the gateway for the real servers.

NAT Router netmask

If the NAT router's floating IP needs a particular netmask, select it from drop-down list.

NAT Router device

Use this text field to define the device name of the network interface for the floating IP address, such as **`eth1:1`**.

Tip

You should alias the NAT floating IP address to the Ethernet interface connected to the private network. In this example, the private network is on the `eth1` interface, so **`eth1:1`** is the floating IP address.

Warning

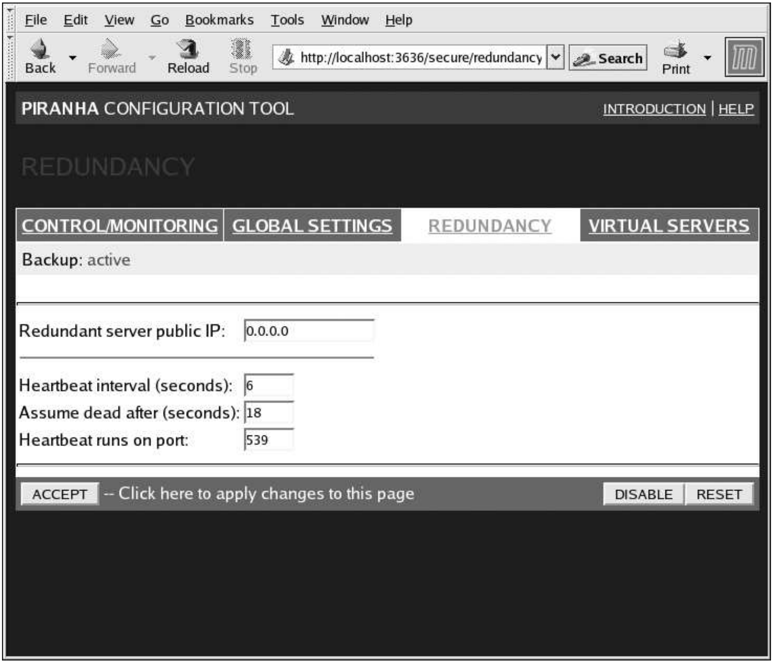
After completing this page, click the **ACCEPT** button to make sure you do not lose any changes when selecting a new panel.

10.5. REDUNDANCY

The **REDUNDANCY** panel allows you to configure of the backup LVS router node and set various heartbeat monitoring options.

Tip

The first time you visit this screen, it displays an "inactive" **Backup** status and an **ENABLE** button. To configure the backup LVS router, click on the **ENABLE** button so that the screen matches Figure 10-4.


File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://localhost:3636/secure/redundancy
Search
Print
PIRANHA CONFIGURATION TOOL
INTRODUCTION | HELP
REDUNDANCY
CONTROL/MONITORING
GLOBAL SETTINGS
REDUNDANCY
VIRTUAL SERVERS
ACCEPT
-- Click here to apply changes to this page
DISABLE
RESET

Backup: active

Redundant server public IP: 0.0.0.0

Heartbeat interval (seconds): 6

Assume dead after (seconds): 18

Heartbeat runs on port: 539

Figure 10-4. The REDUNDANCY Panel

Redundant server public IP

Enter the public real IP address for the backup LVS router node.

Redundant server private IP

Enter the backup node's private real IP address in this text field.

If you do not see the field called **Redundant server private IP**, go back to the **GLOBAL SETTINGS** panel and enter a **Primary server private IP** address and click **ACCEPT**.

The rest of the panel is devoted to configuring the heartbeat channel, which is used by the backup node to monitor the primary node for failure.

Heartbeat Interval (seconds)

This field sets the number of seconds between heartbeats — the interval that the backup node will check the functional status of the primary LVS node.

Assume dead after (seconds)

If the primary LVS node does not respond after this number of seconds, then the backup LVS router node will initiate failover.

Heartbeat runs on port

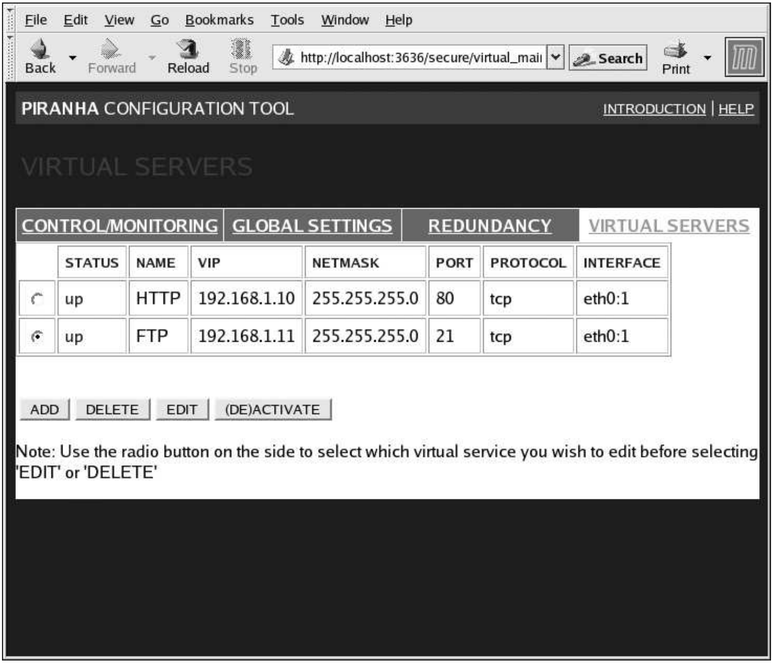
This field sets the port at which the heartbeat communicates with the primary LVS node. The default is set to 539 if this field is left blank.

Warning

Remember to click the **ACCEPT** button after making any changes in this panel to make sure you do not lose any changes when selecting a new panel.

10.6. VIRTUAL SERVERS

The **VIRTUAL SERVERS** panel displays information for each currently defined virtual server. Each table entry shows the status of the virtual server, the server name, the virtual IP assigned to the server, the netmask of the virtual IP, the port number to which the service communicates, the protocol used, and the virtual device interface.



	STATUS	NAME	VIP	NETMASK	PORT	PROTOCOL	INTERFACE
○	up	HTTP	192.168.1.10	255.255.255.0	80	tcp	eth0:1
●	up	FTP	192.168.1.11	255.255.255.0	21	tcp	eth0:1

Note: Use the radio button on the side to select which virtual service you wish to edit before selecting 'EDIT' or 'DELETE'

Figure 10-5. The VIRTUAL SERVERS Panel

Each server displayed in the **VIRTUAL SERVERS** panel can be configured on subsequent screens or *subsections*.

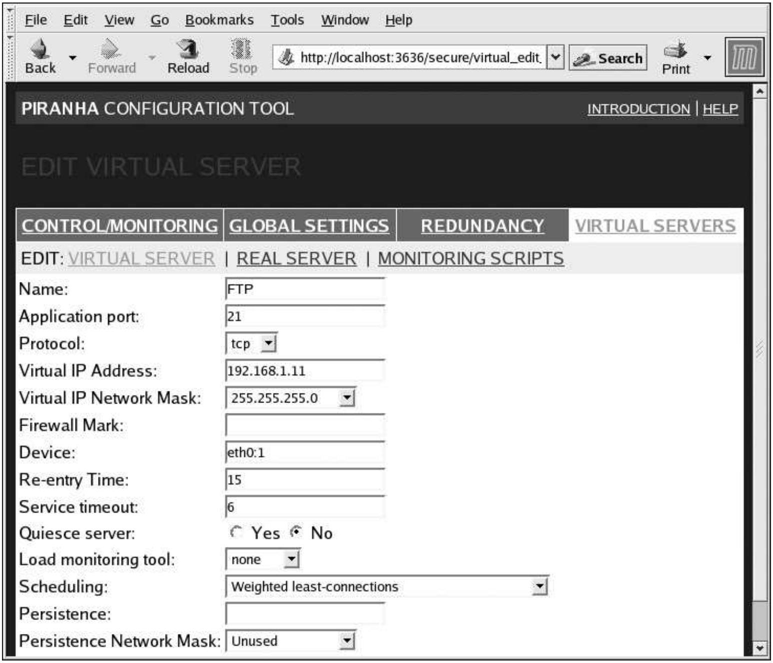
To add a service, click the **ADD** button. To remove a service, select it by clicking the radio button next to the virtual server and click the **DELETE** button.

To enable or disable a virtual server in the table click its radio button and click the **(DE)ACTIVATE** button.

After adding a virtual server, you can configure it by clicking the radio button to its left and clicking the **EDIT** button to display the **VIRTUAL SERVER** subsection.

10.6.1. The VIRTUAL SERVER Subsection

The **VIRTUAL SERVER** subsection panel shown in Figure 10-6 allows you to configure an individual virtual server. Links to subsections related specifically to this virtual server are located along the top of the page. But before configuring any of the subsections related to this virtual server, complete this page and click on the **ACCEPT** button.



Name:	FTP
Application port:	21
Protocol:	tcp
Virtual IP Address:	192.168.1.11
Virtual IP Network Mask:	255.255.255.0
Firewall Mark:	
Device:	eth0:1
Re-entry Time:	15
Service timeout:	6
Quiesce server:	○ Yes ● No
Load monitoring tool:	none
Scheduling:	Weighted least-connections
Persistence:	
Persistence Network Mask:	Unused

Figure 10-6. The VIRTUAL SERVERS Subsection

Name

Enter a descriptive name to identify the virtual server. This name is *not* the hostname for the machine, so make it descriptive and easily identifiable. You can even reference the protocol used by the virtual server, such as HTTP.

Application port

Enter the port number through which the service application will listen. Since this example is for HTTP services, port 80 is used.

Protocol

Choose between UDP and TCP in the drop-down menu. Web servers typically communicate via the TCP protocol, so this is selected in the example above.

Virtual IP Address

Enter the virtual server's floating IP address in this text field.

Virtual IP Network Mask

Set the netmask for this virtual server with the drop-down menu.

Firewall Mark

Do *not* enter a firewall mark integer value in this field unless you are bundling multi-port protocols or creating a multi-port virtual server for separate, but related protocols. In this example, the above virtual server has a **Firewall Mark** of 80 because we are bundling connections to HTTP on port 80 and to HTTPS on port 443 using the firewall mark value of 80. When combined with persistence, this technique will ensure users accessing both insecure and secure webpages are routed to the same real server, preserving state.

> ⚠ **Warning**
>
> Entering a firewall mark in this field allows IPVS to recognize that packets bearing this firewall mark are treated the same, but you must perform further configuration outside of the **Piranha Configuration Tool** to actually assign the firewall marks. See Section 9.3 *Multi-port Services and LVS Clustering* for instructions on creating multi-port services and Section 9.4 *FTP In an LVS Cluster* for creating a highly available FTP virtual server.

Device

Enter the name of the network device to which you want the floating IP address defined the **Virtual IP Address** field to bind.

You should alias the public floating IP address to the Ethernet interface connected to the public network. In this example, the public network is on the `eth0` interface, so **`eth0:1`** should be entered as the device name.

Re-entry Time

Enter an integer value which defines the length of time, in seconds, before the active LVS router attempts to bring a real server back into the cluster after a failure.

Service Timeout

Enter an integer value which defines the length of time, in seconds, before a real server is considered dead and removed from the cluster.

Quiesce server

When the **Quiesce server** radio button is selected, anytime a new real server node comes online, the least-connections table is reset to zero so the active LVS router routes requests as if all the real servers were freshly added to the cluster. This option prevents the a new server from becoming bogged down with a high number of connections upon entering the cluster.

Load monitoring tool

The LVS router can monitor the load on the various real servers by using either `rup` or `ruptime`. If you select `rup` from the drop-down menu, each real server must run the `rstatd` service. If you select `ruptime`, each real server must run the `rwhod` service.

Caution

Load monitoring is *not* the same as load balancing and can result in hard to predict scheduling behavior when combined with weighted scheduling algorithms. Also, if you use load monitoring, the real servers in the cluster must be Linux machines.

Scheduling

Select your preferred scheduling algorithm from the drop-down menu. The default is **`Weighted least-connection`**. For more information on scheduling algorithms, see Section 7.3.1 *Scheduling Algorithms*.

Persistence

If an administrator needs persistent connections to the virtual server during client transactions, enter the number of seconds of inactivity allowed to lapse before a connection times out in this text field.

Important

If you entered a value in the **Firewall Mark** field above, you should enter a value for persistence as well. Also, be sure that if you use firewall marks and persistence together, that the amount of persistence is the same for each virtual server with the firewall mark. For more on persistence and firewall marks, refer to Section 7.5 *Persistence and Firewall Marks*.

Persistence Network Mask

To limit persistence to particular subnet, select the appropriate network mask from the drop-down menu.

Note

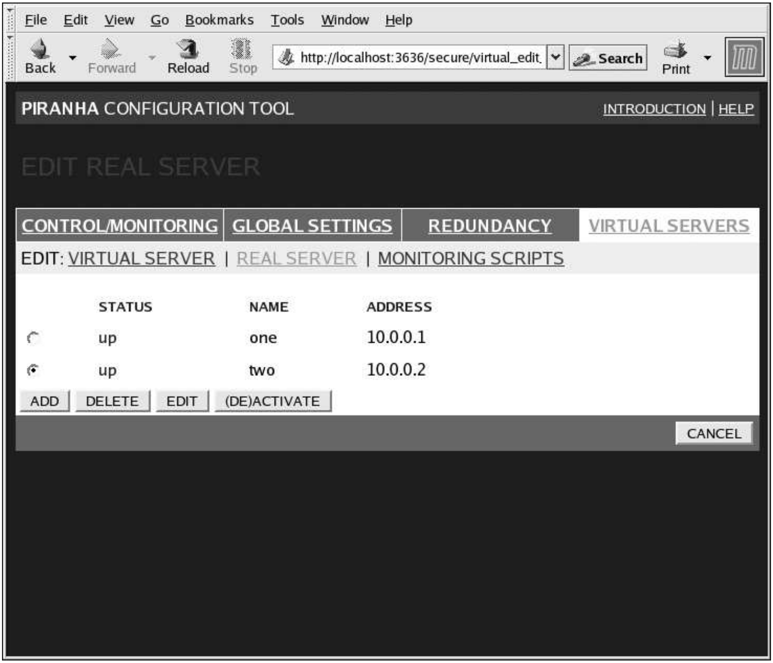
Before the advent of firewall marks, persistence limited by subnet was a crude way of bundling connections. Now, it is best to use persistence in relation to firewall marks to achieve the same result.

Warning

Remember to click the **ACCEPT** button after making any changes in this panel. To make sure you do not lose changes when selecting a new panel.

10.6.2. REAL SERVER Subsection

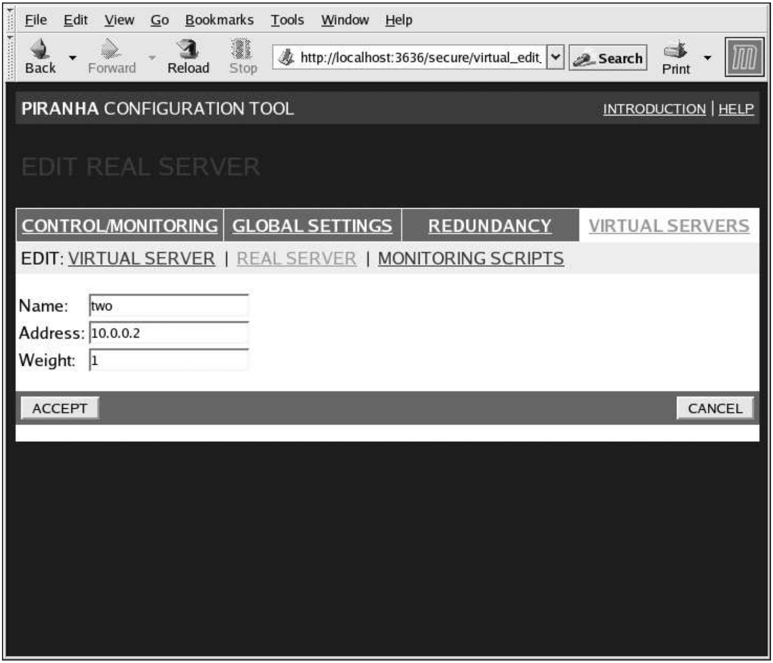
Clicking on the **REAL SERVER** subsection link at the top of the panel displays the **EDIT REAL SERVER** subsection. It displays the status of the physical server hosts for a particular virtual service.


File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://localhost:3636/secure/virtual_edit_
Search
Print
PIRANHA CONFIGURATION TOOL
INTRODUCTION | HELP
EDIT REAL SERVER
CONTROL/MONITORING
GLOBAL SETTINGS
REDUNDANCY
VIRTUAL SERVERS
EDIT: VIRTUAL SERVER | REAL SERVER | MONITORING SCRIPTS
ADD
DELETE
EDIT
(DE)ACTIVATE
CANCEL

	STATUS	NAME	ADDRESS
○	up	one	10.0.0.1
●	up	two	10.0.0.2

Figure 10-7. The REAL SERVER Subsection

Click the **ADD** button to add a new server. To delete an existing server, select the radio button beside it and click the **DELETE** button. Click the **EDIT** button to load the **EDIT REAL SERVER** panel, as seen in Figure 10-8.


File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://localhost:3636/secure/virtual_edit_
Search
Print
PIRANHA CONFIGURATION TOOL
INTRODUCTION | HELP
EDIT REAL SERVER
CONTROL/MONITORING
GLOBAL SETTINGS
REDUNDANCY
VIRTUAL SERVERS
EDIT: VIRTUAL SERVER | REAL SERVER | MONITORING SCRIPTS
ACCEPT
CANCEL

Name: two
Address: 10.0.0.2
Weight: 1

Figure 10-8. The REAL SERVER Configuration Panel

This panel consists of three entry fields:

Name

A descriptive name for the real server.

Tip

This name is *not* the hostname for the machine, so make it descriptive and easily identifiable.

Address

The real server's IP address. Since the listening port is already specified for the associated virtual server, do not add a port number.

Weight

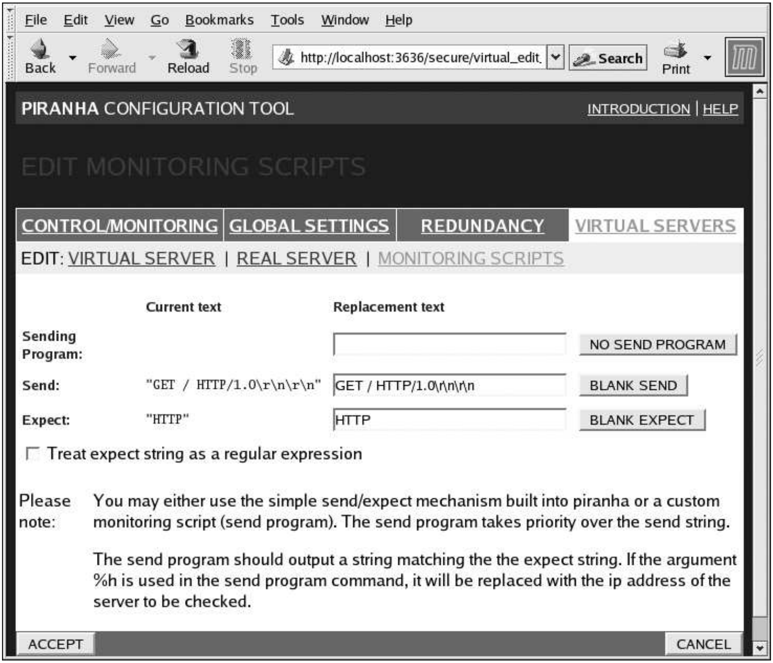
An integer value indicating this host's capacity relative to that of other hosts in the pool. The value can be arbitrary, but treat it as a ratio in relation to other real servers in the cluster. For more on server weight, see Section 7.3.2 *Server Weight and Scheduling*.

Warning

Remember to click the **ACCEPT** button after making any changes in this panel. To make sure you do not lose any changes when selecting a new panel.

10.6.3. EDIT MONITORING SCRIPTS Subsection

Click on the **MONITORING SCRIPTS** link at the top of the page. The **EDIT MONITORING SCRIPTS** subsection allows the administrator to specify a send/expect string sequence to verify that the service for the virtual server is functional on each real server. It is also the place where the administrator can specify customized scripts to check services requiring dynamically changing data.



	Current text	Replacement text
Sending Program:		
Send:	"GET / HTTP/1.0\r\n\r\n"	GET / HTTP/1.0\r\n\r\n
Expect:	"HTTP"	HTTP

☐ Treat expect string as a regular expression

Please note: You may either use the simple send/expect mechanism built into piranha or a custom monitoring script (send program). The send program takes priority over the send string.

The send program should output a string matching the the expect string. If the argument %h is used in the send program command, it will be replaced with the ip address of the server to be checked.

Figure 10-9. The EDIT MONITORING SCRIPTS Subsection

Sending Program

For more advanced service verification, you can use this field to specify the path to a service-checking script. This functionality is especially helpful for services that require dynamically changing data, such as HTTPS or SSL.

To use this functionality, you must write a script that returns a textual response, set it to be executable, and type the path to it in the **Sending Program** field.

Tip

To ensure that each server in the real server pool is checked, use the special token `%h` after the path to the script in the **Sending Program** field. This token is replaced with each real server's IP address as the script is called by the `nanny` daemon.

The following is a sample script to use as a guide when composing an external service-checking script:

```
#!/bin/sh

TEST=‘dig -t soa example.com @$1 | grep -c dns.example.com

if [ $TEST != "1" ]; then
 echo "OK
else
 echo "FAIL"
fi
```

Note

If an external program is entered in the **Sending Program** field, then the **Send** field is ignored.

Send

Enter a string for the `nanny` daemon to send to each real server in this field. By default the send field is completed for HTTP. You can alter this value depending on your needs. If you leave this field blank, the `nanny` daemon attempts to open the port and assume the service is running if it succeeds.

Only one send sequence is allowed in this field, and it can only contain printable, ASCII characters as well as the following escape characters:

- \n for new line.
- \r for carriage return.
- \t for tab.
- \ to escape the next character which follows it.

Expect

Enter a the textual response the server should return if it is functioning properly. If you wrote your own sending program, enter the response you told it to send if it was successful.

Tip

To determine what to send for a given service, you can open a `telnet` connection to the port on a real server and see what is returned. For instance, FTP reports 220 upon connecting, so could enter **`quit`** in the **Send** field and **`220`** in the **Expect** field.

Warning

Remember to click the **ACCEPT** button after making any changes in this panel. To make sure you do not lose any changes when selecting a new panel.

Once you have configured virtual servers using the **Piranha Configuration Tool**, you must copy specific configuration files to the backup LVS router. See Section 10.7 *Synchronizing Configuration Files* for details.

10.7. Synchronizing Configuration Files

After configuring the primary LVS router, there are several configuration files that must be copied to the backup LVS router before you start the cluster.

These files include:

- `/etc/sysconfig/ha/lvs.cf` — the configuration file for the LVS routers.
- `/etc/sysctl` — the configuration file that, among other things, turns on packet forwarding in the kernel.
- `/etc/sysconfig/iptables` — If you are using firewall marks, you should synchronize one of these files based on which network packet filter you are using.

Important

The `/etc/sysctl.conf` and `/etc/sysconfig/iptables` files do *not* change when you configure the cluster using the **Piranha Configuration Tool**.

10.7.1. Synchronizing `lvs.cf`

Anytime the LVS configuration file, `/etc/sysconfig/ha/lvs.cf`, is created or updated, you must copy it to the backup LVS router node.

Warning

Both the active and backup LVS router nodes must have identical `lvs.cf` files. Mismatched LVS configuration files between the LVS router nodes can prevent failover.

The best way to do this is to use the `scp` command.

☆ **Important**

> To use `scp` the `sshd` must be running on the backup router, see Section 8.1 *Configuring Services on the LVS Routers* for details on how to properly configure the necessary services on the LVS routers.

Issue the following command as the root user from the primary LVS router to sync the **`lvs.cf`** files between the router nodes:

```
scp /etc/sysconfig/ha/lvs.cf n.n.n.n:/etc/sysconfig/ha/lvs.cf
```

In the above command, replace *`n.n.n.n`* with the real IP address of the backup LVS router.

10.7.2. Synchronizing `sysctl`

The `sysctl` file is only modified once in most situations. This file is read at boot time and tells the kernel to turn on packet forwarding.

☆ **Important**

> If you are not sure whether or not packet forwarding is enabled in the kernel, see Section 8.5 *Turning on Packet Forwarding* for instructions on how to check and, if necessary, enable this key functionality.

10.7.3. Synchronizing Network Packet Filtering Rules

If you are using `iptables`, you will need to synchronize the appropriate configuration file on the backup LVS router.

If you alter the any network packet filter rules, enter the following command as root from the primary LVS router:

```
scp /etc/sysconfig/iptables n.n.n.n:/etc/sysconfig/
```

In the above command, replace *`n.n.n.n`* with the real IP address of the backup LVS router.

Next either open an ssh session to the backup router or log into the machine as root and type the following command:

```
/sbin/service iptables restart
```

Once you have copied these files over to the backup router and started the appropriate services (see Section 8.1 *Configuring Services on the LVS Routers* for more on this topic) you are ready to start the cluster.

10.8. Starting the Cluster

To start the LVS cluster, it is best to have two root terminals open simultaneously or two simultaneous root open ssh sessions to the primary LVS router.

In one terminal, watch the kernel log messages with the command:

```
tail -f /var/log/messages
```

Then start the cluster by typing the following command into the other terminal:

```
/sbin/service pulse start
```

Follow the progress of the pulse service's startup in the terminal with the kernel log messages. When you see the following output, the pulse daemon has started properly:

```
gratuitous lvs arps finished
```

To stop watching /var/log/messages, type [Ctrl]-[c].

From this point on, the primary LVS router is also the active LVS router. While you can make requests to the cluster at this point, you should start the backup LVS router before putting the cluster into service. To do this, simply repeat the process described above on the backup LVS router node.

After completing this final step, the cluster will be up and running.

III. Appendixes

Table of Contents

Appendix A.

Supplementary Hardware Information

The following sections provide additional information about configuring the hardware used in a cluster system.

A.1. Attached Storage Requirements

The following sections detail the steps to consider when directly connecting storage devices to cluster nodes, whether using SCSI host-bus adapters or Fibre Channel connections.

A.2. Setting Up a Fibre Channel Interconnect

Fibre Channel can be used in either single-initiator or multi-initiator configurations.

A single-initiator Fibre Channel interconnect has only one node connected to it. This may provide better host isolation and better performance than a multi-initiator bus. Single-initiator interconnects ensure that each node is protected from disruptions due to the workload, initialization, or repair of the other node.

If employing a RAID array that has multiple host ports, and the RAID array provides simultaneous access to all the shared logical units from the host ports on the storage enclosure, set up single-initiator Fibre Channel interconnects to connect each node to the RAID array. If a logical unit can fail over from one controller to the other, the process must be transparent to the operating system.

Figure A-1 shows a single-controller RAID array with two host ports and the host bus adapters connected directly to the RAID controller, without using Fibre Channel hubs or switches. When using this type of single-initiator Fibre Channel connection, your RAID controller must have a separate host port for each cluster node.

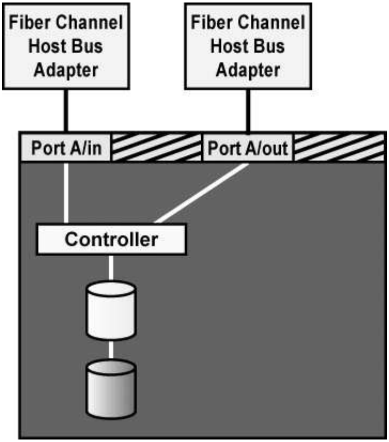

Fiber Channel Host Bus Adapter
Fiber Channel Host Bus Adapter
Port A/in
Port A/out
Controller

Figure A-1. Single-controller RAID Array Connected to Single-initiator Fibre Channel Interconnects

The external RAID array must have a separate SCSI channel for each cluster node. In clusters with more than two nodes, connect each node to the SCSI channel on the RAID array, using a single-initiator SCSI bus as shown in Figure A-1.

To connect multiple cluster nodes to the same host port on the RAID array, use a Fibre Channel hub or switch. In this case, each host bus adapter is connected to the hub or switch, and the hub or switch is connected to a host port on the RAID controller.

A Fibre Channel hub or switch is also required with a dual-controller RAID array with two host ports on each controller. This configuration is shown in Figure A-2. Additional cluster nodes may be connected to either Fibre Channel hub or switch shown in the diagram. Some RAID arrays include a built-in hub so that each host port is already connected to each of the internal RAID controllers. In this case, an additional external hub or switch may not be needed.

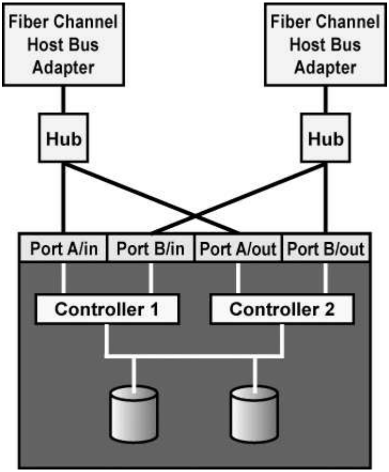


Figure A-2. Dual-controller RAID Array Connected to Single-initiator Fibre Channel Interconnects

A.3. SCSI Storage Requirements

A single-initiator SCSI bus has only one node connected to it, and provides host isolation and better performance than a multi-initiator bus. Single-initiator buses ensure that each node is protected from disruptions due to the workload, initialization, or repair of the other nodes.

When using a single- or dual-controller RAID array that has multiple host ports and provides simultaneous access to all the shared logical units from the host ports on the storage enclosure, the setup of the single-initiator SCSI buses to connect each cluster node to the RAID array is possible. If a logical unit can fail over from one controller to the other, the process must be transparent to the operating system. Note that some RAID controllers restrict a set of disks to a specific controller or port. In this case, single-initiator bus setups are not possible.

A single-initiator bus must adhere to the requirements described in Section A.3.1 *SCSI Configuration Requirements*.

To set up a single-initiator SCSI bus configuration, perform the following steps:

- Enable the onboard termination for each host bus adapter.
- Enable the termination for each RAID controller.

- Use the appropriate SCSI cable to connect each host bus adapter to the storage enclosure.

Setting host bus adapter termination is done in the adapter BIOS utility during system boot. To set RAID controller termination, refer to the vendor documentation. Figure A-3 shows a configuration that uses two single-initiator SCSI buses.

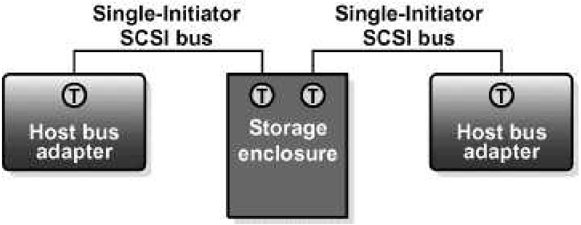


Figure A-3. Single-initiator SCSI Bus Configuration

Figure A-4 shows the termination in a single-controller RAID array connected to two single-initiator SCSI buses.

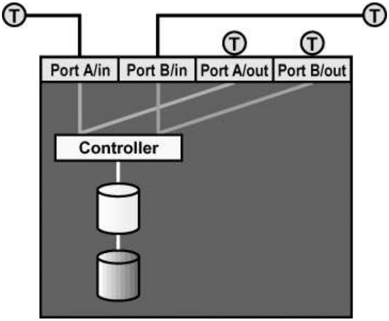


Figure A-4. Single-controller RAID Array Connected to Single-initiator SCSI Buses

Figure A-5 shows the termination in a dual-controller RAID array connected to two single-initiator SCSI buses.

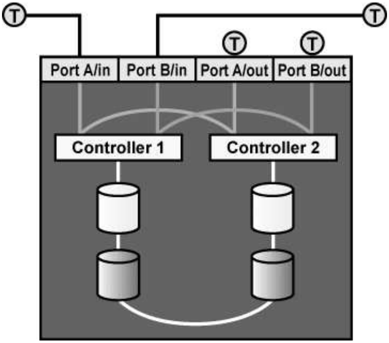


Figure A-5. Dual-controller RAID Array Connected to Single-initiator SCSI Buses

A.3.1. SCSI Configuration Requirements

SCSI devices must adhere to a number of configuration requirements to operate correctly. Failure to adhere to these requirements adversely affects cluster operation and resource availability.

The following is an overview of SCSI configuration requirements:

- Buses must be terminated at each end. Refer to Section A.3.2 *SCSI Bus Termination* for more information.
- Buses must not extend beyond the maximum length restriction for the bus type. Internal cabling must be included in the length of the SCSI bus. Refer to Section A.3.3 *SCSI Bus Length* for more information.
- All devices (host bus adapters and disks) on a bus must have unique SCSI identification numbers. Refer to Section A.3.4 *SCSI Identification Numbers* for more information.
- The Linux device name for each shared SCSI device must be the same on each cluster system. For example, a device named `/dev/sdc` on one cluster system must be named `/dev/sdc` on the other cluster system. One way to ensure that devices are named the same is by using identical hardware for both cluster systems.

Use the system's configuration utility to set SCSI identification numbers and enable host bus adapter termination. When the system boots, a message is displayed describing how to start the utility. For example, the utility prompts the user to press [Ctrl]-[A], and follow the prompts to perform a particular task. To set storage enclosure and RAID controller termination, refer to the vendor documentation. Refer to Section A.3.2 *SCSI Bus Termination* and Section A.3.4 *SCSI Identification Numbers* for more information.

A.3.2. SCSI Bus Termination

A SCSI bus is an electrical path between two terminators. A device (host bus adapter, RAID controller, or disk) attaches to a SCSI bus by a short *stub*, which is an unterminated bus segment that usually must be less than 0.1 meter in length.

Buses must have only two terminators located at opposing ends of the bus. Additional terminators, terminators that are not at the ends of the bus, or long stubs cause the bus to operate incorrectly. Termination for a SCSI bus can be provided by the devices connected to the bus or by external terminators, if the internal (onboard) device termination can be disabled.

Testing has shown that external termination on HBAs that run at speeds greater than 80MB/second does not work reliably.

When disconnecting a device from a single-initiator SCSI bus follow these guidelines:

- Unterminated SCSI cables must not be connected to an operational host bus adapter or storage device.
- Connector pins must not bend or touch an electrical conductor while the SCSI cable is disconnected.
- To disconnect a host bus adapter from a single-initiator bus, first disconnect the SCSI cable from the RAID controller and then from the adapter. This ensures that the RAID controller is not exposed to any erroneous input.
- Protect connector pins from electrostatic discharge while the SCSI cable is disconnected by wearing a grounded anti-static wrist guard and physically protecting the cable ends from contact with other objects.
- Do not remove a device that is currently participating in any SCSI bus transactions.

To enable or disable an adapter's internal termination, use the system BIOS utility. When the system boots, a message is displayed describing how to start the utility. For example, many utilities prompt users to press [Ctrl]-[A]. Follow the prompts for setting the termination. At this point, it is also possible to set the SCSI identification number, as needed, and disable SCSI bus resets. Refer to Section A.3.4 *SCSI Identification Numbers* for more information.

To set storage enclosure and RAID controller termination, refer to the vendor documentation.

A.3.3. SCSI Bus Length

A SCSI bus must adhere to length restrictions for the bus type. Buses that do not adhere to these restrictions do not operate properly. The length of a SCSI bus is calculated from one terminated end to the other and must include any cabling that exists inside the system or storage enclosures.

A cluster supports LVD (low voltage differential) buses. The maximum length of a single-initiator LVD bus is 25 meters. The maximum length of a multi-initiator LVD bus is 12 meters. According to the SCSI standard, a single-initiator LVD bus is a bus that is connected to only two devices, each within 0.1 meter from a terminator. All other buses are defined as multi-initiator buses.

Do not connect any single-ended devices to an LVD bus; doing so converts the bus single-ended, which has a much shorter maximum length than a differential bus.

A.3.4. SCSI Identification Numbers

Each device on a SCSI bus must have a unique SCSI identification number. Devices include host bus adapters, RAID controllers, and disks.

The number of devices on a SCSI bus depends on the data path for the bus. A cluster supports wide SCSI buses, which have a 16-bit data path and support a maximum of 16 devices. Therefore, there are sixteen possible SCSI identification numbers that can be assigned to the devices on a bus.

In addition, SCSI identification numbers are prioritized. Use the following priority order to assign SCSI identification numbers:

```
7 - 6 - 5 - 4 - 3 - 2 - 1 - 0 - 15 - 14 - 13 - 12 - 11 - 10 - 9 - 8
```

The previous order specifies that 7 is the highest priority, and 8 is the lowest priority. The default SCSI identification number for a host bus adapter is 7, because adapters are usually assigned the highest priority. It is possible to assign identification numbers for logical units in a RAID subsystem by using the RAID management interface.

To modify an adapter's SCSI identification number, use the system BIOS utility. When the system boots, a message is displayed describing how to start the utility. For example, a user may be prompted to press [Ctrl]-[A] and follow the prompts for setting the SCSI identification number. At this point, it is possible to enable or disable the adapter's internal termination, as needed, and disable SCSI bus resets. Refer to Section A.3.2 *SCSI Bus Termination* for more information.

The prioritized arbitration scheme on a SCSI bus can result in low-priority devices being locked out for some period of time. This may cause commands to time out, if a low-priority storage device, such as a disk, is unable to win arbitration and complete a command that a host has queued to it. For some workloads, it is possible to avoid this problem by assigning low-priority SCSI identification numbers to the host bus adapters.

Appendix B.

Selectively Installing Red Hat Cluster Suite Packages

B.1. Installing the Red Hat Cluster Suite Packages

Red Hat Cluster Suite consists of the following RPM packages:

- `rgmanager` — Manages cluster services and resources
- `system-config-cluster` — Contains the **Cluster Configuration Tool**, used to graphically configure the cluster and the display of the current status of the nodes, resources, fencing agents, and cluster services
- `ccsd` — Contains the cluster configuration services daemon (`ccsd`) and associated files
- `magma` — Contains an interface library for cluster lock management
- `magma-plugins` — Contains plugins for the `magma` library
- `cman` — Contains the Cluster Manager (CMAN), which is used for managing cluster membership, messaging, and notification
- `cman-kernel` — Contains required CMAN kernel modules
- `dlm` — Contains distributed lock management (DLM) library
- `dlm-kernel` — Contains required DLM kernel modules
- `fence` — The cluster I/O fencing system that allows cluster nodes to connect to a variety of network power switches, fibre channel switches, and integrated power management interfaces
- `gulm` — Contains the GULM lock management userspace tools and libraries (an alternative to using CMAN and DLM).
- `iddev` — Contains libraries used to identify the file system (or volume manager) in which a device is formatted

Also, you can optionally install Red Hat GFS on your Red Hat Cluster Suite. Red Hat GFS consists of the following RPMs:

- `GFS` — The Red Hat GFS module
- `GFS-kernel` — The Red Hat GFS kernel module
- `gnbd` — The GFS Network Block Device module

- `gnbd-kernel` — Kernel module for the GFS Network Block Device
- `lvm2-cluster` — Cluster extensions for the logical volume manager
- `GFS-kernheaders` — GFS kernel header files
- `gnbd-kernheaders` — `gnbd` kernel header files

Tip

> You can access the Red Hat Cluster Suite and Red Hat GFS products by using Red Hat Network to subscribe to and access the channels containing the Red Hat Cluster Suite and Red Hat GFS packages. From the Red Hat Network channel, you can manage entitlements for your cluster nodes and upgrade packages for each node within the Red Hat Network Web-based interface. For more information on using Red Hat Network, visit the following URL:
>
> `http://rhn.redhat.com`

You can install Red Hat Cluster Suite and Red Hat GFS RPMs using either of the following methods:

- Automatic RPM installation — Using `up2date`
- Custom RPM installation — Selectively installing RPMs using the `rpm` utility

For automatic RPM installation, refer to Section B.1.1 *Automatic RPM Installation*. For custom RPM installation, refer to Section B.1.2 *Custom RPM Installation*.

B.1.1. Automatic RPM Installation

Automatic RPM installation consists of running the `up2date` utility at each node for the Red Hat Cluster Suite and Red Hat GFS products.

Note

> If you are installing the GFS RPMs, you must run `up2date` for Red Hat Cluster Suite before running it for Red Hat GFS.

To automatically install RPMs, do the following at each node:

1. Log on as the root user.

2. Run `up2date --installall --channel` *`Label`* for Red Hat Cluster Suite. The following example shows running the command for i386 RPMs:

   ```
   # up2date --installall --channel rhel-i386-as-4-cluster
   ```

3. (Optional) If you are installing Red Hat GFS, run `up2date --installall --channel` *`Label`* for Red Hat GFS. The following example shows running the command for i386 RPMs:

   ```
   # up2date --installall --channel rhel-i386-as-4-gfs-6.1
   ```

B.1.2. Custom RPM Installation

Custom RPM installation consists of the following steps:

1. Determine which RPMs to install. For information on determining which RPMs to install, refer to Section B.1.2.1 *Determining RPMs To Install*.
2. Install the RPMs using the `rpm` utility. For information about installing the RPMs using the `rpm` utility, refer to Section B.1.2.2 *Installing Packages with the `rpm` Utility*.

Note

If you are installing the GFS RPMs, you must install Red Hat Cluster Suite before Red Hat GFS.

B.1.2.1. Determining RPMs To Install

Determining which RPMs to install is based on the following criteria:

- The lock manager Red Hat Cluster Suite is using — either DLM or GULM
- The Red Hat Cluster Suite and Red Hat GFS functions you are using (besides the standard functions)
- Whether to include development libraries
- The type of kernel (or kernels) is installed

Use the following tables for determining which RPMs to install:

- Table B-1 — For Red Hat Cluster Suite with DLM
- Table B-2 — For Red Hat Cluster Suite with GULM

- Table B-3 — For Red Hat GFS

The tables contain the following information to assist you in determining which packages to install:

- RPMs — The names of the RPMs (excluding revision numbers)
- Inclusion — The tables provide the following information about whether an RPM should be included in the installation:
  - Req: Required RPM — You *must* install the RPM.
  - Opt: Optional RPM — Refer to the "Purpose" for more information about determining whether to include the RPM.
  - Dev: Development RPM — Used for development purposes. Refer to the "Purpose" for more information about determining whether to include the RPM.
- Purpose — Provides a concise description of the RPM purpose. Assists in determining which RPMs to include other than the required RPMs.

To determine which RPMs to include in the installation, perform the following steps:

1. Determine whether you are installing Red Hat Cluster Suite with DLM or Red Hat Cluster Suite with GULM.
   a. If you are installing Red Hat Cluster Suite with DLM, refer to Table B-1 to identify which RPMs are required, optional, and for development.
   b. If you are installing Red Hat Cluster Suite with GULM, refer to Table B-2 to identify which RPMs are required, optional, and for development.
2. If you are installing Red Hat GFS, refer to Table B-3 to identify which RPMs are required, optional, and for development.
3. With the information gathered in the previous steps, proceed to install the RPMs using the procedures in Section B.1.2.2 *Installing Packages with the `rpm` Utility*.

RPMs	Inclusion	Depends on Kernel Type?	Purpose
`ccs-ver-rel.arch`	Req	No	The Cluster Configuration System
`cman-ver-rel.arch`	Req	No	The Cluster Manager

`cman-kernel-ver-rel.arch`
`cman-kernel-hugemem-ver-rel.arch`
`cman-kernel-smp-ver-rel.arch`
Note: The types of RPMs available vary according to RHN channel.
	Req	Yes	The Cluster Manager kernel modules

`dlm-ver-rel.arch`	Req	No	The Distributed Lock Manager

`dlm-kernel-ver-rel.arch`
`dlm-kernel-hugemem-ver-rel.arch`
`dlm-kernel-smp-ver-rel.arch`
Note: The types of RPMs available vary according to RHN channel.
	Req	Yes	The Distributed Lock Manager kernel modules

`fence-ver-rel.arch`	Req	No	The cluster I/O fencing system
`iddev-ver-rel.arch`	Req	No	A library that identifies device contents
`magma-ver-rel.arch`	Req	No	A cluster/lock manager API abstraction library
`magma-plugins-ver-rel.arch`	Req	No	Cluster manager plugins for magma
`gulm-ver-rel.arch` *Note:* The `gulm` module is required with DLM because the `magma-plugins` module has a dependency on the `gulm` RPM.	Req	No	The Grand Unified Lock Manager (GULM, available for this release and earlier versions of Red Hat GFS)
`perl-Net-Telnet-ver-rel.arch`	Req	No	Net-Telnet Perl module

RPMs	Inclusion	Depends on Kernel Type?	Purpose
rgmanager-*ver-rel.arch*	Opt	No	Open source HA resource group failover
system-config-cluster-*ver-rel.arch*	Req	No	GUI to manage cluster configuration
ipvsadm-*ver-rel.arch*	Opt	No	Utility to administer the Linux Virtual Server
piranha-*ver-rel.arch*	Opt	No	Cluster administration tools
ccs-devel-*ver-rel.arch*	Dev	No	CCS static library
cman-kernheaders-*ver-rel.arch*	Dev	No	`cman` kernel header files
dlm-devel-*ver-rel.arch*	Dev	No	The Distributed Lock Manager user-space libraries
dlm-kernheaders-*ver-rel.arch*	Dev	No	`dlm` kernel header files
iddev-devel-*ver-rel.arch*	Dev	No	`iddev` development libraries
magma-devel-*ver-rel.arch*	Dev	No	A cluster/lock manager API abstraction library

Table B-1. RPM Selection Criteria: Red Hat Cluster Suite with DLM

RPMs	Inclusion	Depends on Kernel Type?	Purpose
`ccs-*ver*-*rel*.*arch*`	Req	No	The Cluster Configuration System
`fence-*ver*-*rel*.*arch*`	Req	No	The cluster I/O fencing system
`gulm-*ver*-*rel*.*arch*`	Req	No	The Grand Unified Lock Manager (GULM, available for this release and earlier versions of Red Hat GFS)
`iddev-*ver*-*rel*.*arch*`	Req	No	A library that identifies device contents
`magma-*ver*-*rel*.*arch*`	Req	No	A cluster/lock manager API abstraction library
`magma-plugins-*ver*-*rel*.*arch*`	Req	No	Cluster manager plugins for magma
`perl-Net-Telnet-*ver*-*rel*.*arch*`	Req	No	Net-Telnet Perl module
`system-config-cluster-*ver*-*rel*.*arch*`	Req	No	GUI to manage cluster configuration
`ipvsadm-*ver*-*rel*.*arch*`	Opt	No	Utility to administer the Linux Virtual Server
`piranha-*ver*-*rel*.*arch*`	Opt	No	Cluster administration tools
`ccs-devel-*ver*-*rel*.*arch*`	Dev	No	CCS static library
`gulm-devel-*ver*-*rel*.*arch*`	Dev	No	`gulm` libraries
`iddev-devel-*ver*-*rel*.*arch*`	Dev	No	`iddev` development libraries

RPMs	Inclusion	Depends on Kernel Type?	Purpose
`magma-devel-`*`ver-rel.arch`*	Dev	No	A cluster/lock manager API abstraction library

Table B-2. RPM Selection Criteria: Red Hat Cluster Suite with GULM

RPMs	Inclusion	Depends on Kernel Type?	Purpose
`GFS-`*`ver-rel.arch`*	Req	No	The Red Hat GFS module

`GFS-kernel-`*`ver-rel.arch`*
`GFS-kernel-hugemem-`*`ver-rel.arch`*
`GFS-kernel-smp-`*`ver-rel.arch`*
Note: The types of RPMs available vary according to RHN channel.
	Req	Yes	The Red Hat GFS kernel modules

`gnbd-`*`ver-rel.arch`*	Opt	No	The GFS Network Block Device

`gnbd-kernel-`*`ver-rel.arch`*
`gnbd-kernel-hugemem-`*`ver-rel.arch`*
`gnbd-kernel-smp-`*`ver-rel.arch`*
Note: The types of RPMs available vary according to RHN channel.
	Opt	Yes	Kernel module for GFS Network Block Device

`lvm2-cluster-`*`ver-rel.arch`*	Req	No	Cluster extensions for the logical volume manager
`GFS-kernheaders-`*`ver-rel.arch`*	Dev	No	GFS kernel header files
`gnbd-kernheaders-`*`ver-rel.arch`*	Dev	No	`gnbd` kernel header files

Table B-3. RPM Selection Criteria: Red Hat GFS

B.1.2.2. Installing Packages with the `rpm` Utility

You can use the `rpm` utility to install RPMs from CDs created with RHN ISOs. The procedure consists of copying RPMs to a local computer, removing the RPMs that are not needed for the installation, copying the RPMs to the cluster nodes, and installing them.

To install the RPMs, follow these instructions:

1. At a local computer (one that is not part of the cluster) make a temporary directory to contain the RPMs. For example:
   ```
   $ mkdir /tmp/RPMS/
   ```
2. Insert the Red Hat Cluster Suite CD into the CD-ROM drive.

   Note

   If a **Question** dialog box is displayed that asks if you want to run `autorun`, click **No**.

3. Copy all the RPM files from the CD (located in `/media/cdrom/RedHat/RPMS/`) to the temporary directory created earlier. For example:
   ```
   $ cp /media/cdrom/RedHat/RPMS/*.rpm  /tmp/RPMS/
   ```

   Note

   If your local computer is running a version of Red Hat Enterprise Linux that is earlier than Red Hat Enterprise Linux 4, the path to the RPMs on the CD may be different. For example, on Red Hat Enterprise Linux 3, the path is `/mnt/cdrom/RedHat/RPMS/`.

4. Eject the CD from the CD-ROM drive.
5. (Optional) If you are installing Red Hat GFS, insert a Red Hat GFS CD into the CD-ROM drive. If you are not installing Red Hat GFS, proceed to step 8.

   Note

   If a **Question** dialog box is displayed that asks if you want to run `autorun`, click **No**.

6. Copy all the RPM files from the CD (located in `/media/cdrom/RedHat/RPMS/`) to the temporary directory created earlier. For example:
   ```
   $ cp /media/cdrom/RedHat/RPMS/*.rpm  /tmp/RPMS/
   ```

Note

> If your local computer is running a version of Red Hat Enterprise Linux that is earlier than Red Hat Enterprise Linux 4, the path to the RPMs on the CD may be different. For example, on Red Hat Enterprise Linux 3, the path is `/mnt/cdrom/RedHat/RPMS/`.

7. Eject the CD from the CD-ROM drive.
8. Change to the temporary directory containing the copied RPM files. For example:
   ```
   $ cd /tmp/RPMS/
   ```
9. Remove the "-kernel" RPMs for kernels that are not installed in the cluster node, and any other RPMs that are not being installed (for example, optional or development RPMS). The following example removes SMP and hugemem "-kernel" RPM files:
   ```
   $ rm *-kernel-smp* *-kernel-hugemem*
   ```
   For information about selecting the RPMs to install, refer to Section B.1.2.1 *Determining RPMs To Install*.
10. Log in to each cluster node as the root user and make a directory to contain the RPMs. For example:
    ```
    # mkdir /tmp/node-RPMS/
    ```
11. Copy the RPMs from the temporary directory in the local computer to directories in the cluster nodes using the `scp` command. For example, to copy the RPMs to node rhcs-node-01, run the following command at the local computer:
    ```
    $ scp /tmp/RPMS/*.rpm root@rhcs-node-01:/tmp/node-RPMS/
    ```
12. At each node (logged in as root), change to the temporary directory created earlier (`/tmp/node-RPMS`) and install the RPMs by running the `rpm` utility as follows:
    ```
    # cd /tmp/node-RPMS/
    # rpm -Uvh *
    ```

Appendix C.

Multipath-usage.txt File for Red Hat Enterprise Linux 4 Update 3

This appendix contains the `Multipath-usage.txt` file. The file is included with the `dm-multipath` RPM and provides guidelines for using `dm-multipath` with Red Hat Cluster Suite for Red Hat Enterprise Linux 4 Update 3:

```
    RHEL4 U3 Device Mapper Multipath Usage


Overview
------------
Device Mapper Multipath (DM-MP) allows nodes to route I/O over
multiple paths to a storage controller. A path refers to the
connection from an HBA port to a storage controller port. As paths
fail and new paths come up, DM-MP reroutes the I/O over the
available paths.

When there are multiple paths to a storage controller, each path
appears as a separate device.  DM-MP creates a new device on top of
those devices. For example, a node with two HBAs attached to a storage
controller with two ports via a single unzoned FC switch sees four
devices: /dev/sda, /dev/sdb, /dev/sdc, and /dev/sdd. DM-MP creates a
single device, /dev/mpath/mpath1 that reroutes I/O to those four
underlying devices.

DM-MP consists of the following components:

o dm-multipath kernel module -- This module reroutes I/O and fails
  over paths and path groups.

o multipath command -- This command configures, lists, and removes
  multipath  devices. The command is run in rc.sysinit during startup,
   and by udev, whenever a block device    is added.

o multipathd daemon -- This daemon monitors paths, checking to see
  if faulty paths have been fixed. As paths come back up, multipathd
  may also initiate path group switches to ensure that the optima
  path group is being used. Also, it is possible to interactively
  modify a multipath device.

o kpartx command -- This command creates Device Mapper devices for the
  partitions on a device. It is necessary to use this command for DOS-
  based partitions with DM-MP.
```

DM-MP works with a variety of storage arrays. It auto-configures the following storage arrays:

o 3PARdata VV
o Compaq HSV110
o Compaq MSA1000
o DDN SAN DataDirector
o DEC HSG80
o EMC SYMMETRIX
o EMC CLARiiON
o FSC CentricStor
o HITACHI DF400
o HITACHI DF500
o HITACHI DF600
o HP HSV110
o HP HSV210
o HP A6189A
o HP Open-
o IBM 3542
o IBM ProFibre 4000R
o NETAPP
o SGI TP9100
o SGI TP9300
o SGI TP9400
o SGI TP9500
o STK OPENstroage D280
o SUN StorEdge 3510
o SUN T4

Storage arrays not included in the list may require entries in the /etc/multipath.conf file.

NOTE: Some storage arrays require special handling of I/O errors and path-group switching. Those require separate hardware handler kernel modules.

Terms and Concepts

Hardware Handler:
A kernel module that performs hardware-specific actions when switching path groups and dealing with I/O errors.

Path:
The connection from an HBA port to a storage controller port for a LUN. Each path appears as a separate device. Paths can be in various states (refer to "Path States").

```
Path States:
    ready  - Path is able to handle I/O requests.
    shaky  - Path is up, but temporarily not available for normal
             operations.
    faulty - Path is unable to handle I/O requests.
    ghost  - Path is a passive path, on an active/passive
             controller.

        NOTE: The shaky and ghost states only exist for certain
              storage arrays.

Path Group:
    A grouping of paths. With DM-MP, only one path group--the
    active path group--receives I/O at any time. Within a path
    group, DM-MP selects which ready path should receive I/O in
    a round robin fashion. Path groups can be in various states
    (refer to "Path Group States").

Path Group States:
    active   - Path group currently receiving I/O requests.
    enabled  - Path groups to try if the active path group has no paths
               in the ready state.
    disabled - Path groups to try if the active path group and all
           enabled path groups have no paths in the active state.

        NOTE: The disabled state only exists for certain storage arrays.

Path Priority:
    Each path can have a priority assigned to it by a callout program.
    Path priorities can be used to group paths by priority and change
    their relative weights for the round robin path selector.

Path Group Priority:
    Each path group has a priority that is equal to the sum of the
    priorities of all the non-faulty paths in the group. By default, the
    multipathd daemon tries to ensure that the path group with the
    highest priority is always in the active state.

Failover:
    When I/O to a path fails, the dm-multipath module tries to switch to
    an enabled path group. If there are no enabled path groups with
    any paths in the ready state, dm-multipath tries to switch to a
    disabled path group. If necessary, dm-multipath runs the hardware
    handler for the multipath device.

Failback:
    At regular intervals, multipathd checks the current priority of
    all path groups. If the current path group is not the highest
    priority path group, multipathd reacts according to the failback
    mode. By default, multipathd immediately switches to the highest
```

priority path group. Other options for multipathd are to (a) wait for a user-defined length of time (for the path groups to stabilize) and then switch or (b) for multipathd to do nothing and wait for manual intervention. Failback can be forced at any time by running the multipath command.

Multipath device:

The multipath device is the device mapper device created by dm-multipath. A multipath device can be identified by either its WWID or its alias. A multipath device has one or more path groups. It also has numerous attributes defined in the following file:

/usr/share/doc/device-mapper-multipathd-0.4.5/multipath.conf.annotated

alias:

The alias is the name of a multipath device. By default, the alias is set to the WWID. However, by setting the "user_friendly_names" configuration option, the alias is set to a unique name of the form mpath<n>. The alias name can also be explicitly set for each multipath device in the configuration file.

NOTE: While the alias in guaranteed to be unique on a node, it is not guaranteed to be the same on all nodes using the multipath device. Also, it may change.

WWID:

The WWID (World Wide Identifier) is an identifier for the multipath device that is guaranteed to be globally unique and unchanging. It is determined by the getuid callout program.

Using DM-MP

Initial setup:

1. If it is not already installed. Install the device-mapper-multipath package.

2. Edit /etc/multipath.conf. For new installations, all devices are blacklisted. The default blacklist is listed in the commented out section of /etc/multipath.conf. If you comment out or delete the following lines in /etc/multipath.conf, the default blacklist takes effect:

```
devnode_blacklist {
        devnode "*"
}
```

For some conditions, that may not be sufficient. If DM-MP is multipathing devices that you do not want it to work on, you can blacklist the devices by either device name or WWID.

NOTE: It is safest to blacklist individual devices by WWID, because their device names may change.

Several other configuration options are detailed later in this document. To check the effects of configuration changes, you can do a dry run with the following command:

```
# multipath -v2 -d
```

3. Set the multipathd init script to run at boot time. by issuing the commands

```
# chkconfig --add multipathd
# chkconfig multipathd on
```

4. start dm-multipath (This is only necessary the first time. On reboot, this should happen automatically).

```
# multipath
# /etc/init.d/multipathd start
```

After initial setup, all access to the multipathed storage should go through the multipath device.

Configuration File:

Many features of DM-MP are configurable using the configuration file, /etc/multipath.conf.

For a complete list of all options with descriptions, refer to /usr/share/doc/device-mapper-multipathd-0.4.5/multipath.conf.annotated

The configuration file is divided into four sections: system defaults, blacklisted devices (devnode_blacklist), per storage array model settings (devices), and per multipath device settings (multipaths). The per multipath device settings are used for the multipath device with a matching "wwid" value. The per storage array model settings are used for all multipath devices with matching "vendor" and "product" values. To determine the attributes of a multipath device, first the per multipath settings are checked, then the per controller settings, then the system defaults. The blacklisted device section is described setup step 2.

NOTE: There are compiled-in defaults for the "defaults",

```
      "devnode_blacklist", and "devices" sections of the
      configuration file. To see what these are, refer to the
      following file:

/usr/share/doc/device-mapper-multipathd-0.4.5/multipath.conf.synthetic

If you are using one of the storage arrays listed in the preceding
text (in "Overview"), you probably do not need to modify the "devices"
subsection. If you are using a simple disk enclosure, the defaults
should work. If you are using a storage array that is not
listed, you may need to create a "devices" subsection for your array.

Explanation of output
-----------------------
When you create, modify, or list a multipath device, you get a
printout of the current device setup. The format is as follows.

For each multipath device:

action_if_any: alias (wwid_if_different_from_alias)
[size][features][hardware_handler]

For each path group:

\_ scheduling_policy [path_group_priority_if_known]
[path_group_status_if_known]


For each path:

 \_ host:channel:id:lun devnode major:minor [path_status]
 [dm_status_if_known]


NOTE: The preceding lines for path group and path
      were broken because of print limitations.

The dm status (dm_status_if_known) is like the path status
(path_status), but from the kernel's point of view.  The dm status
has two states: "failed", which is analogous to "faulty",
and "active" which covers all other path states. Occasionally,
the path state and the dm state of a device will temporarily
not agree.

NOTE: When a multipath device is being created or modified, the
path group status and the dm status are not known.  Also, the
features are not always correct. When a multipath device is being
isted, the path group priority is not known.

Restrictions
---------------
```

```
DM-MP cannot be run on either the root or boot device.

Other Sources of information
----------------------------
Configuration file explanation:
/usr/share/doc/device-mapper-multipathd-0.4.5/multipath.conf.annotated

Upstream documentation:
http://christophe.varoqui.free.fr/wiki/wakka.php?wiki=Home

mailing list:
dm-devel@redhat.com
Subscribe to this from https://www.redhat.com/mailman/listinfo/dm-devel.
The list archives are at https://www.redhat.com/archives/dm-devel/

Man pages:
multipath.8, multipathd.8, kpartx.8 mpath_ctl.8
```

Index

Colophon

The manuals are written in DocBook SGML v4.1 format. The HTML and PDF formats are produced using custom DSSSL stylesheets and custom jade wrapper scripts. The DocBook SGML files are written using **Emacs** with the help of PSGML mode.

Garrett LeSage created the admonition graphics (note, tip, important, caution, and warning). They may be freely redistributed with the Red Hat documentation.

The Red Hat Product Documentation Team consists of the following people:

Sandra A. Moore — Primary Writer/Maintainer of the *Red Hat Enterprise Linux Installation Guide for x86, Itanium™, AMD64, and Intel® Extended Memory 64 Technology (Intel® EM64T)*; Primary Writer/Maintainer of the *Red Hat Enterprise Linux Installation Guide for the IBM® POWER Architecture*; Primary Writer/Maintainer of the *Red Hat Enterprise Linux Installation Guide for the IBM® S/390® and IBM® eServer™ zSeries® Architectures*

John Ha — Primary Writer/Maintainer of the *Red Hat Cluster Suite Configuring and Managing a Cluster*; Co-writer/Co-maintainer of the *Red Hat Enterprise Linux Security Guide*; Maintainer of custom DocBook stylesheets and scripts

Edward C. Bailey — Primary Writer/Maintainer of the *Red Hat Enterprise Linux Introduction to System Administration*; Primary Writer/Maintainer of the *Release Notes*; Contributing Writer to the *Red Hat Enterprise Linux Installation Guide for x86, Itanium™, AMD64, and Intel® Extended Memory 64 Technology (Intel® EM64T)*

Karsten Wade — Primary Writer/Maintainer of the *Red Hat SELinux Guide*; Contributing Writer to the *Red Hat Enterprise Linux System Administration Guide*

Andrius T. Benokraitis — Primary Writer/Maintainer of the *Red Hat Enterprise Linux Reference Guide*; Co-writer/Co-maintainer of the *Red Hat Enterprise Linux Security Guide*; Contributing Writer to the *Red Hat Enterprise Linux System Administration Guide*

Paul Kennedy — Primary Writer/Maintainer of the *Red Hat GFS Administrator's Guide*; Contributing Writer to the *Red Hat Cluster Suite Configuring and Managing a Cluster*

Mark Johnson — Primary Writer/Maintainer of the *Red Hat Desktop Deployment Guide*; Contributing Writer of Red Hat Network documentation

Melissa Goldin — Primary Writer/Maintainer of the *Red Hat Enterprise Linux Step By Step Guide*; Contributing Writer of Red Hat Network Documentation

Lucy Ringland — Red Hat Enterprise Linux Documentation Editor.

The Red Hat Localization Team consists of the following people:

Amanpreet Singh Alam — Punjabi translations

Jean-Paul Aubry — French translations

David Barzilay — Brazilian Portuguese translations

Runa Bhattacharjee — Bengali translations

Chester Cheng — Traditional Chinese translations

Verena Fuehrer — German translations

Kiyoto Hashida — Japanese translations

N. Jayaradha — Tamil translations

Michelle Jiyeen Kim — Korean translations

Yelitza Louze — Spanish translations

Noriko Mizumoto — Japanese translations

Ankitkumar Rameshchandra Patel — Gujarati translations

Rajesh Ranjan — Hindi translations

Nadine Richter — German translations

Audrey Simons — French translations

Francesco Valente — Italian translations

Sarah Wang — Simplified Chinese translations

Ben Hung-Pin Wu — Traditional Chinese translations

Tongjie Tony Fu — Simplified Chinese Translations

Manuel Ospina — Spanish Translations