Red Hat Enterprise Linux 5

Sinopsis de la suite para Cluster

Red Hat Cluster Suite para Red Hat Enterprise Linux 5

Edición 3

Last Updated: 2018-04-17

Red Hat Enterprise Linux 5 Sinopsis de la suite para Cluster

Red Hat Cluster Suite para Red Hat Enterprise Linux 5
Edición 3

Landmann
rlandmann@redhat.com

Legal Notice

Resumen

La Sinopsis de Red Hat Cluster Suite proporciona un resumen de Red Hat Cluster Suite para Red Hat Enterprise Linux 5

Table of Contents

INTRODUCCIÓN

Este documento proporciona un resumen de Red Hat Cluster Suite para Red Hat Enterprise Linux 5 y está organizado de la siguiente manera:

- Capítulo 1, *Sinopsis de Red Hat Cluster Suite*
- Capítulo 2, *Resumen de componentes de Red Hat Cluster Suite*

Aunque este documento es un resumen, el lector debe tener un conocimiento avanzado de Red Hat Enterprise Linux y entender los conceptos alrededor de la computación de servidores con el fin de poder asimilar la información dada.

Para obtener mayor información sobre Red Hat Enterprise Linux, consulte los siguientes recursos:

- *Manual de instalación de Red Hat Enterprise Linux*— Proporciona información relacionada con la instalación de Red Hat Enterprise Linux 5.
- *Manual de implementación de Red Hat Enterprise Linux*— Proporciona información sobre la implementación, configuración y administración de Red Hat Enterprise Linux 5.

Para obtener mayor información sobre Red Hat Cluster Suite para Red Hat Enterprise Linux 5, consulte los siguientes recursos:

- *Configuración y administración de Red Hat Cluster*— Proporciona información sobre la instalación, configuración y administración de componentes de Red Hat Cluster.
- *LVM Administrator's Guide: Configuration and Administration*— Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- *Sistema de archivos global: Configuración y administración*— Proporciona información sobre la instalación, administración y mantenimiento de Sistemas de archivos global de Red Hat (Red Hat GFS).
- *Sistema de archivos global: Configuración y administración*— Proporciona información sobre la instalación, configuración y mantenimiento de Red Hat GFS2 (Red Hat Global File System 2).
- *Cómo utilizar multirutas de mapeo de dispositivos*— proporciona información sobre el uso de la función multirutas de mapeo de dispositivos de Red Hat Enterprise Linux 5.
- *Como usar GNBD con GFS*— Proporciona información sobre el uso de dispositivos de bloque de red global (GNBD) con GFS de Red Hat.
- *Administración de servidores virtuales en Linux*— Proporciona información sobre cómo configurar sistemas y servicios de alto rendimiento con el servidor virtual de Linux (LVS).
- *Notas de lanzamiento de la Suite Red Hat Cluster*— Proporciona información sobre el lanzamiento actual de Red Hat Cluster Suite.

La documentación para Red Hat Cluster Suite y otros documentos de Red Hat están disponibles en HTML, PDF y RPM en el CD de documentación de Red Hat Enterprise Linux y en http://www.redhat.com/docs/.

1. COMENTARIOS

Si encuentra algún error o si tiene sugerencias para mejorar este documento, nos gustaría escuchar su opinión. Por favor complete un reporte en Bugzilla (http://bugzilla.redhat.com/bugzilla/) usando el componente **Documentation-cluster**.

Be sure to mention the document's identifier:

```
Cluster_Suite_Overview(EN)-5 (2008-12-11T15:49)
```

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

Si tiene una sugerencia para mejorar la documentación trate de ser tan especifico como le sea posible. Si encontró algún error, incluya el número de la sección y parte del texto que rodea el error. Esto ayudará a localizar el error más fácilmente.

CAPÍTULO 1. SINOPSIS DE RED HAT CLUSTER SUITE

Los sistemas de cluster proporcionan fiabilidad, escalabilidad y disponibilidad a servicios de producción crítica. Con Red Hat Cluster Suite, usted puede crear un cluster que cubra sus necesidades de rendimiento, alta disponibilidad, balance de cargas, escalabilidad, compartición de archivos y economía. Este capítulo proporciona un resumen de los componentes y funciones de Red Hat Cluster Suite y consta de las siguientes secciones:

1.1. FUNDAMENTOS SOBRE CLUSTER

Un cluster está compuesto por dos o más computadores (llamados *nodos* o *miembros*) que trabajan juntos para ejecutar una tarea. Hay cuatro clases de cluster:

- Almacenamiento
- Alta disponibilidad
- Balance de carga
- Alto rendimiento

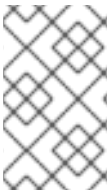
Los cluster de almacenamiento proporcionan una imagen de sistema de archivos consistente a lo largo de los servidores en el cluster, permitiendo que los servidores lean y escriban de forma simultánea a un sistema de archivos compartido. Un cluster de almacenamiento simplifica la administración de almacenamiento al limitar la instalación de aplicaciones a un sistema de archivos. Asimismo, con un sistema de archivos a lo largo del cluster, un cluster de almacenamiento elimina la necesidad de copias de más de los datos de la aplicación y simplifica la creación de copias de seguridad y recuperación contra desastres. Red Hat Cluster Suite proporciona almacenamiento de cluster a través de Red Hat GFS.

Los cluster de alta disponibilidad proporcionan continua disponibilidad de los servicios a través de la eliminación de la falla por un único elemento y a través del proceso de recuperación en contra de fallos al trasladar el servicio desde el nodo de cluster erróneo a otro nodo completamente funcional. Generalmente, los servicios en los cluster de alta disponibilidad leen y escriben datos a través de la lectura y escritura a un sistema de archivos montado. Así, un cluster de alta disponibilidad debe mantener la integridad de los datos cuando un nodo recibe el control del servicio desde otro nodo. Los

nodos erróneos no son vistos por los clientes fuera del cluster. Los cluster de alta disponibilidad son conocidos también como cluster con recuperación contra fallas. Red Hat Cluster Suite proporciona cluster de alta disponibilidad a través del componente de administración de servicios de alta disponibilidad.

Los cluster de balance de carga responden a peticiones de servicios de red desde diferentes nodos para balancear las peticiones a los largo de los nodos del cluster. El balance de carga proporciona escalabilidad económica porque se puede configurar el número de nodos de acuerdo con los requerimientos de balance de carga. Si un nodo en un cluster de balance de carga falla, el software de balance de carga detecta la falla y asigna las peticiones a otros nodos en el cluster. Los nodos erróneos en un cluster de balance de carga no son visibles desde los clientes fuera del cluster. Red Hat Cluster Suite proporciona balance de carga a través de LVS (Servidor Virtual de Linux).

Los cluster de alto rendimiento utilizan los nodos para ejecutar cálculos simultáneos. Un cluster de alto rendimiento permite que las aplicaciones trabajen de forma paralela, mejorando así el rendimiento de éstas. Los cluster de alto rendimiento son conocidos como cluster computacionales o computación de red.



NOTA

Los tipos de cluster resumidos anteriormente reflejan las configuraciones básicas. Según las necesidades del usuario, se podría requerir de una combinación de los cluster descritos.

1.2. RED HAT CLUSTER SUITE INTRODUCTION

Red Hat Cluster Suite (RHCS) es un conjunto integrado de componentes de software que puede ser implementado en una amplia variedad de configuraciones para cubrir las necesidad de rendimiento, alta disponibilidad, balance de carga, escalabilidad, compartición de archivos y economía de recursos.

RHCS consists of the following major components (refer to Figura 1.1, “Red Hat Cluster Suite Introduction”):

- Infraestructura del cluster — proporciona funciones fundamentales para que los nodos trabajen juntos como un cluster: administración del archivo de configuración, administración de membresías, administración de cierres de exclusión y aislamiento.
- Administración de servicios de alta disponibilidad — Proporciona la transferencia de servicios de un cluster a otro en caso de que un nodo falle.
- Herramienta de administración de cluster — Herramientas de configuración y administración para configurar y administrar un cluster de Red Hat. Las herramientas se utilizan con los componentes de infraestructura del cluster, los componentes de alta disponibilidad, de administración de servicios y de almacenamiento.
- Servidor Virtual de Linux (LVS) — Software de encaminamiento que proporciona balance de carga de IP. LVS se ejecuta en un par de servidores pertinentes que distribuyen las peticiones de los clientes a los servidores reales que están tras los servidores LVS.

Se pueden añadir otros componentes a Red Hat Cluster Suite. Estos componentes son parte de un paquete adicional (y *no* parte de Red Hat Cluster Suite):

- Red Hat GFS (Sistema de archivos global) — Proporciona un sistema de archivos de cluster para utilizar con Red Hat Cluster Suite. GFS permite que los nodos compartan el almacenaje a nivel de bloque como si éste fuera conectado localmente en cada nodo.

- Administrador de volúmenes lógicos de cluster (CLVM) — Proporciona administración de volúmenes de almacenamiento en cluster.

NOTA

When you create or modify a CLVM volume for a clustered environment, you must ensure that you are running the **clvmd** daemon. For further information, refer to Sección 1.6, “Administrador de volúmenes lógicos de cluster” .

- Dispositivo de bloque de red global (GNBD) — Un componente complementario de GFS que exporta el almacenaje a nivel de bloque a través de Ethernet. Esta es una manera económica de hacer que el almacenaje de nivel de bloque esté disponible en Red Hat GFS.

For a lower level summary of Red Hat Cluster Suite components and optional software, refer to Capítulo 2, *Resumen de componentes de Red Hat Cluster Suite*

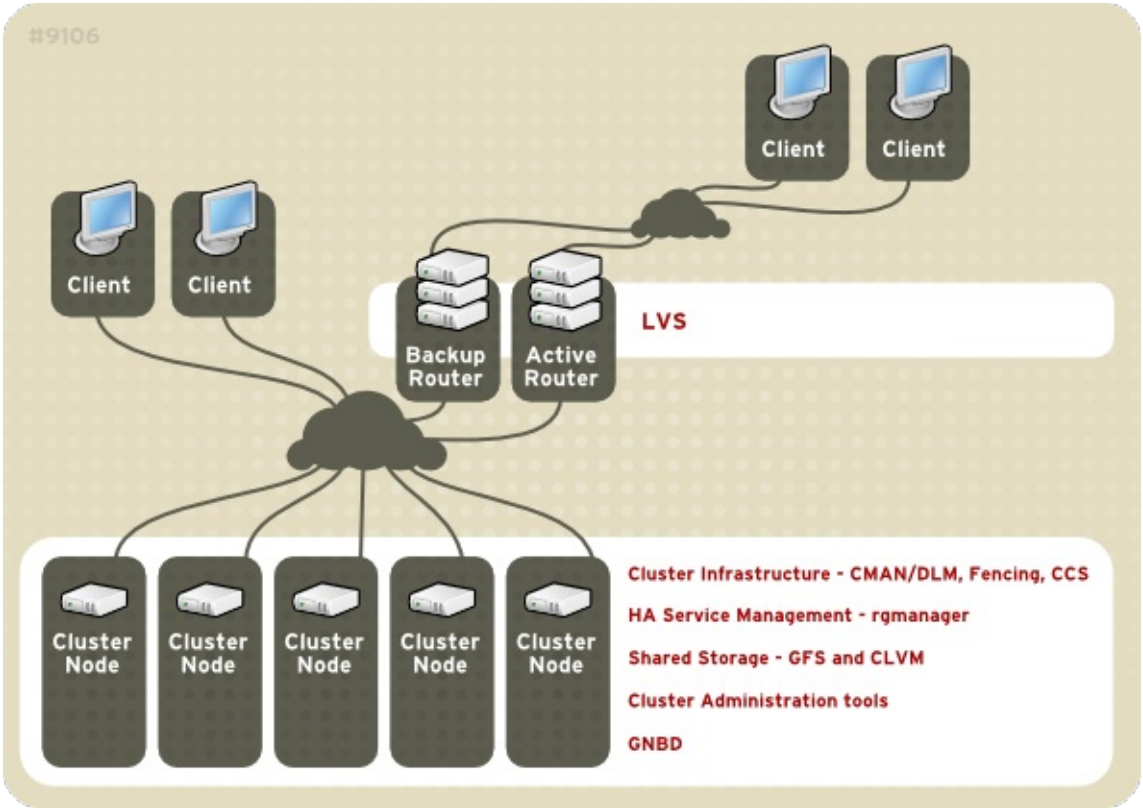


Figura 1.1. Red Hat Cluster Suite Introduction

NOTA

Figura 1.1, “Red Hat Cluster Suite Introduction” includes GFS, CLVM, and GNBD, which are components that are part of an optional package and *not* part of Red Hat Cluster Suite.

1.3. CLUSTER INFRASTRUCTURE

La infraestructura de cluster Red Hat Cluster Suite proporciona las funciones básicas para que un grupo de computadores (llamados *nodos* o *miembros*) trabajen juntos en un cluster. Una vez el cluster ha sido formado con la infraestructura de cluster, se pueden utilizar otros componentes de Red Hat

Cluster Suite para cubrir las necesidades del cluster (por ejemplo, se puede establecer un cluster para compartir archivos en un sistema de archivo GFS o establecer un servicio con recuperación contra fallos). La infraestructura de cluster lleva a cabo las siguientes funciones:

- Administración de cluster
- Administración de los cierres de exclusión
- Fencing
- Administración de la configuración de cluster

1.3.1. Administración del cluster

Cluster management manages cluster quorum and cluster membership. CMAN (an abbreviation for cluster manager) performs cluster management in Red Hat Cluster Suite for Red Hat Enterprise Linux 5. CMAN is a distributed cluster manager and runs in each cluster node; cluster management is distributed across all nodes in the cluster (refer to Figura 1.2, “CMAN/DLM Overview”).

CMAN keeps track of cluster quorum by monitoring the count of cluster nodes. If more than half the nodes are active, the cluster has quorum. If half the nodes (or fewer) are active, the cluster does not have quorum, and all cluster activity is stopped. Cluster quorum prevents the occurrence of a "split-brain" condition — a condition where two instances of the same cluster are running. A split-brain condition would allow each cluster instance to access cluster resources without knowledge of the other cluster instance, resulting in corrupted cluster integrity.

El quórum se determina a través de la comunicación de mensajes entre los nodos del cluster a través de Ethernet. Opcionalmente, se puede determinar el quórum a través de la combinación de mensajes comunicados a través de Ethernet *y* a través de un disco de quórum. Para el quórum a través de Ethernet, el quórum consiste del 50% de los votos más uno. Para el quórum a través del disco de quórum, el quórum depende de las condiciones especificadas por el usuario.

NOTA

Por defecto, cada nodo tiene un voto. Sin embargo, se puede modificar la configuración para que cada nodo tenga más de un voto.

CMAN mantiene rastro de las membresías sondeando los mensajes de otros nodos del cluster. Cuando las membresías del cluster cambian, el administrador de cluster notifica a los otros componentes de la infraestructura para que lleven a cabo las acciones apropiadas. Por ejemplo, cuando el nodo A entra a un cluster y monta el sistema de archivos GFS que los nodos B y C ya tienen montado, se necesita de un nuevo diario y una nueva administración de cierres de exclusión para que el nodo A utilice este sistema de archivos. Si un nodo del cluster no transmite un mensaje durante un tiempo determinado, el administrador del cluster remueve el nodo del cluster y comunica a los otros componentes de la infraestructura de cluster que el nodo no es ya miembro del cluster.

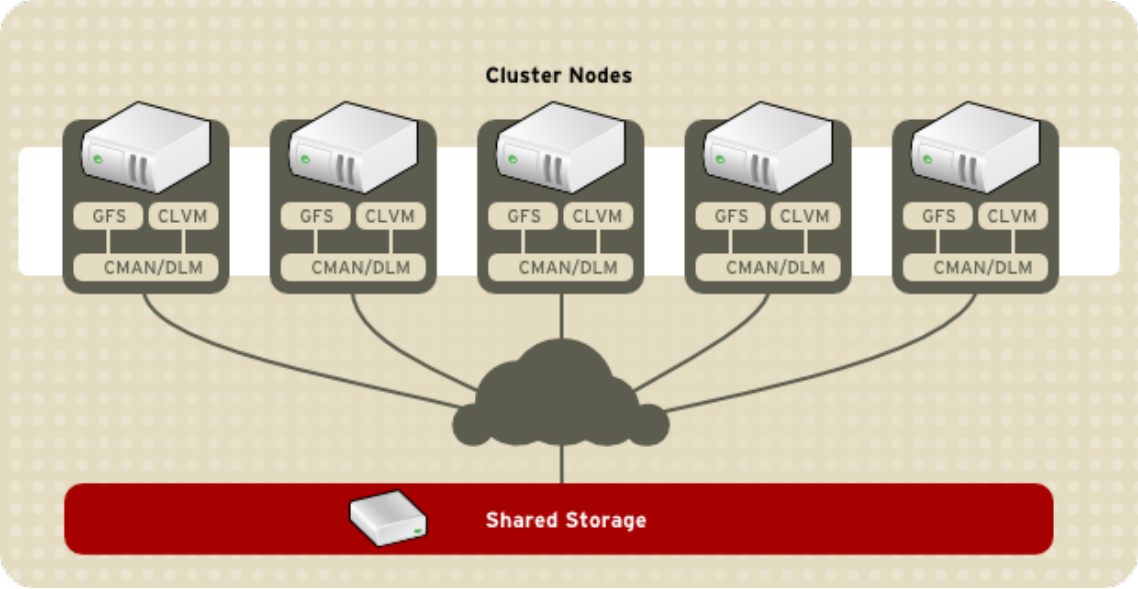


Figura 1.2. CMAN/DLM Overview

1.3.2. Administración de cierres de exclusión

Lock management is a common cluster-infrastructure service that provides a mechanism for other cluster infrastructure components to synchronize their access to shared resources. In a Red Hat cluster, DLM (Distributed Lock Manager) is the lock manager. As implied in its name, DLM is a distributed lock manager and runs in each cluster node; lock management is distributed across all nodes in the cluster (refer to Figura 1.2, “CMAN/DLM Overview”). GFS and CLVM use locks from the lock manager. GFS uses locks from the lock manager to synchronize access to file system metadata (on shared storage). CLVM uses locks from the lock manager to synchronize updates to LVM volumes and volume groups (also on shared storage).

1.3.3. Fencing

Fencing is the disconnection of a node from the cluster's shared storage. Fencing cuts off I/O from shared storage, thus ensuring data integrity. The cluster infrastructure performs fencing through the fence daemon, **fenced**.

Cuando CMAN determina que un nodo ha fallado, CMAN comunica a los otros componentes de la infraestructura de cluster que el nodo ha fallado. **fenced** ejecuta una acción de aislamiento sobre el nodo fallido cuando la comunicación es recibida. Otros componentes de la infraestructura de cluster determinan que acciones se deben tomar — los componentes ejecutan los procedimientos de recuperación que sean necesarios. Por ejemplo, DLM y GFS suspenden sus actividades hasta que detectan que **fenced** ha completado su tarea sobre el nodo fallido. Tras recibir la confirmación de que el nodo ha sido aislado, DLS y GFS ejecutan las tareas de recuperación. DLM libera los cierres del nodo fallido; GFS recupera el registro por diario (journal) del nodo fallido.

El programa de aislamiento determina el método de aislamiento a utilizar desde el archivo de configuración de cluster. Hay dos elementos claves del archivo de configuración de cluster que definen el método de aislamiento: el agente y el dispositivo de aislamiento. El programa de aislamiento hace una llamada al agente de aislamiento especificado en el archivo de configuración del cluster. El agente de aislamiento, a su vez, aisla el nodo a través del dispositivo de aislamiento. Una vez el proceso de aislamiento ha sido completado, el programa de aislamiento notifica al administrador de cluster.

Red Hat Cluster Suite proporciona una variedad de métodos de aislamiento:

- Aislamiento de energía — Un método de aislamiento que utiliza un controlador de energía para apagar el nodo fallido.
- Aislamiento de interruptor de canal de fibra — Un método de aislamiento que desactiva el puerto del canal de fibra que conecta el almacenaje con el nodo fallido.
- GNBD fencing — A fencing method that disables an inoperable node's access to a GNBD server.
- Otros métodos de aislamiento — Hay otros métodos de aislamiento que desactivan la E/S o apagan el nodo fallido. Entre estos se incluye IBM Bladecenters, PAP, DRAC/MC, HP ILO, IPMI, IBM RSA II y otros.

Figura 1.3, “Power Fencing Example” shows an example of power fencing. In the example, the fencing program in node A causes the power controller to power off node D. Figura 1.4, “Fibre Channel Switch Fencing Example” shows an example of Fibre Channel switch fencing. In the example, the fencing program in node A causes the Fibre Channel switch to disable the port for node D, disconnecting node D from storage.

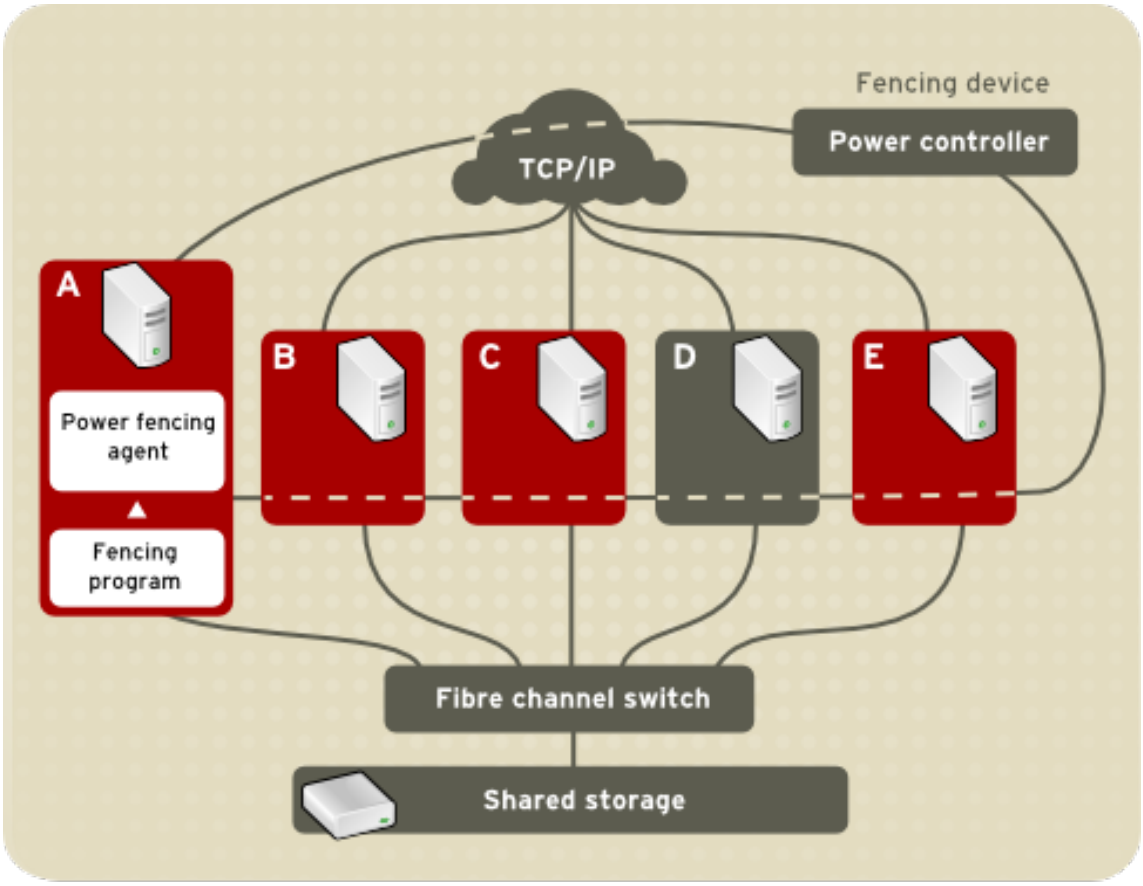


Figura 1.3. Power Fencing Example

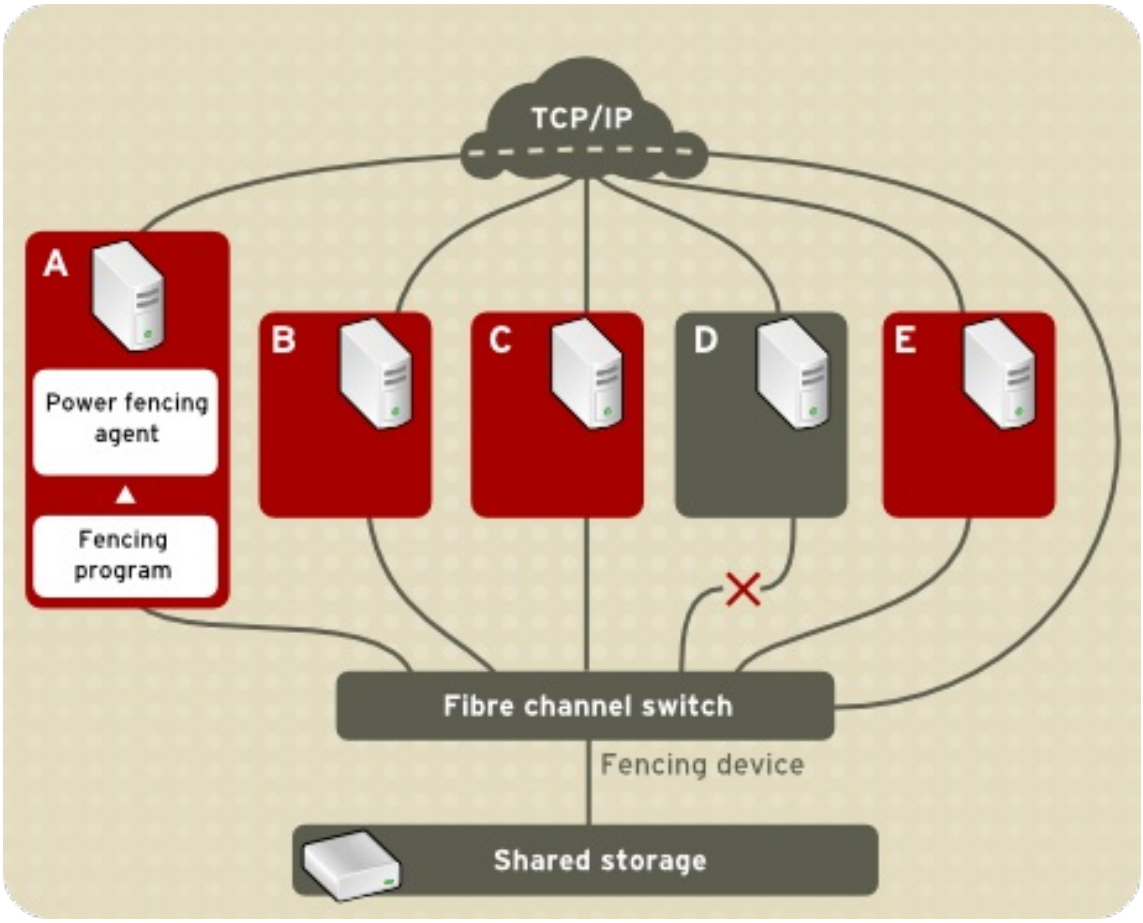


Figura 1.4. Fibre Channel Switch Fencing Example

Para especificar un método de aislamiento se debe editar el archivo de configuración para asignar el nombre del método de aislamiento, el agente de aislamiento y el dispositivo de aislamiento para cada nodo del cluster.

The way in which a fencing method is specified depends on if a node has either dual power supplies or multiple paths to storage. If a node has dual power supplies, then the fencing method for the node must specify at least two fencing devices — one fencing device for each power supply (refer to Figura 1.5, “Fencing a Node with Dual Power Supplies”). Similarly, if a node has multiple paths to Fibre Channel storage, then the fencing method for the node must specify one fencing device for each path to Fibre Channel storage. For example, if a node has two paths to Fibre Channel storage, the fencing method should specify two fencing devices — one for each path to Fibre Channel storage (refer to Figura 1.6, “Fencing a Node with Dual Fibre Channel Connections”).

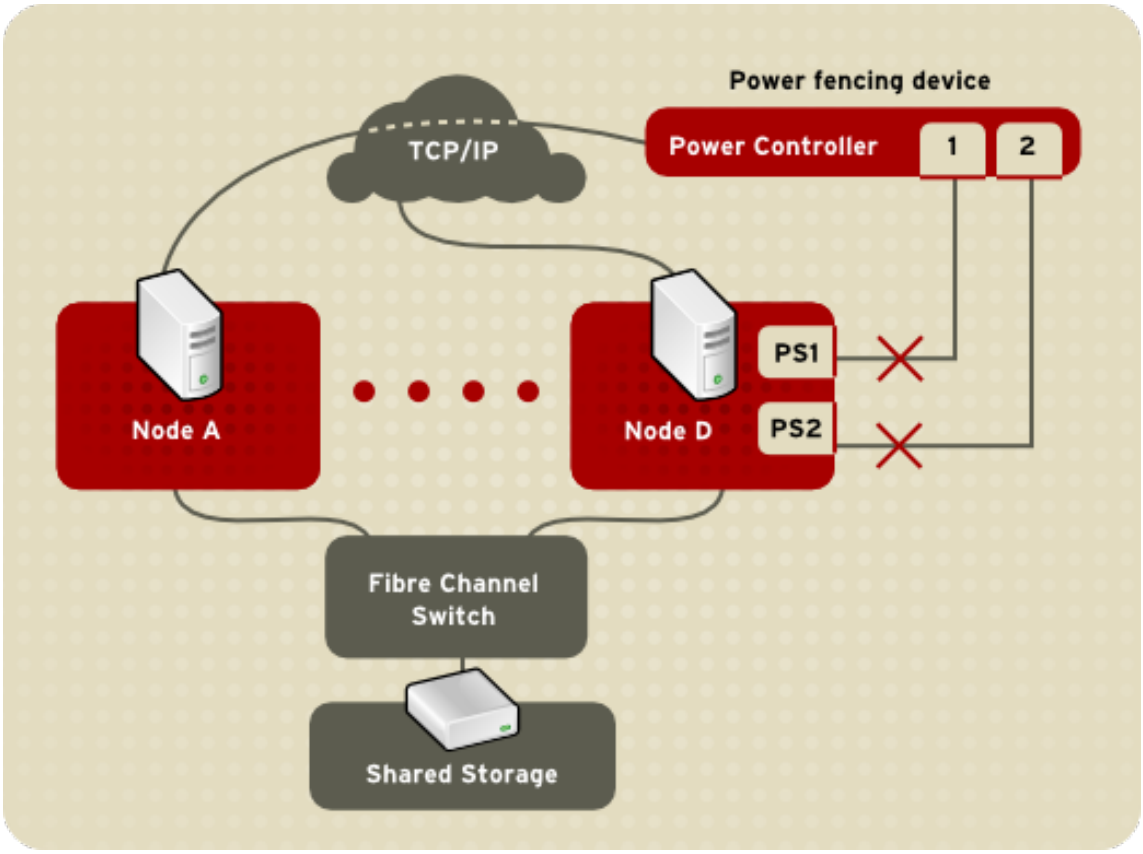


Figura 1.5. Fencing a Node with Dual Power Supplies

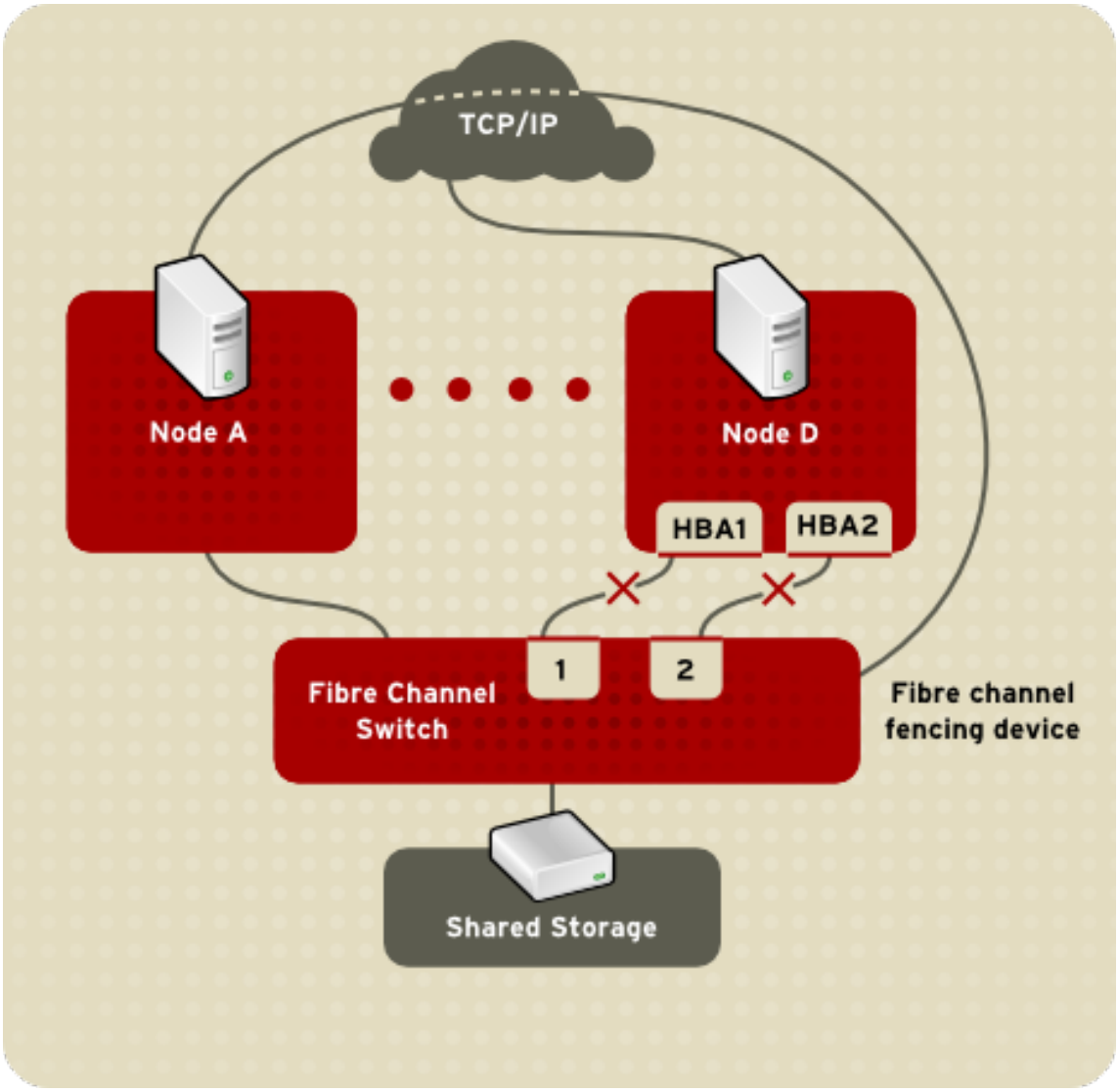


Figura 1.6. Fencing a Node with Dual Fibre Channel Connections

Puede configurar un nodo con uno o más métodos de aislamiento. Cuando se utiliza más de un método de aislamiento, éstos se utilizan en *cascada*, siguiendo el orden de prioridad dado en el archivo de configuración de cluster. Si un nodo falla, éste es aislado mediante el primer método de aislamiento especificado en el archivo de configuración de cluster para ese nodo. Si el primer método no funciona, el siguiente método para ese nodo es utilizado. Si ninguno de los métodos funciona, el primer método de aislamiento es ejecutado de nuevo. Esté bucle continúa hasta que el nodo ha sido aislado satisfactoriamente.

1.3.4. Sistema de configuración del Cluster (CCS)

The Cluster Configuration System (CCS) manages the cluster configuration and provides configuration information to other cluster components in a Red Hat cluster. CCS runs in each cluster node and makes sure that the cluster configuration file in each cluster node is up to date. For example, if a cluster system administrator updates the configuration file in Node A, CCS propagates the update from Node A to the other nodes in the cluster (refer to Figura 1.7, “CCS Overview”).

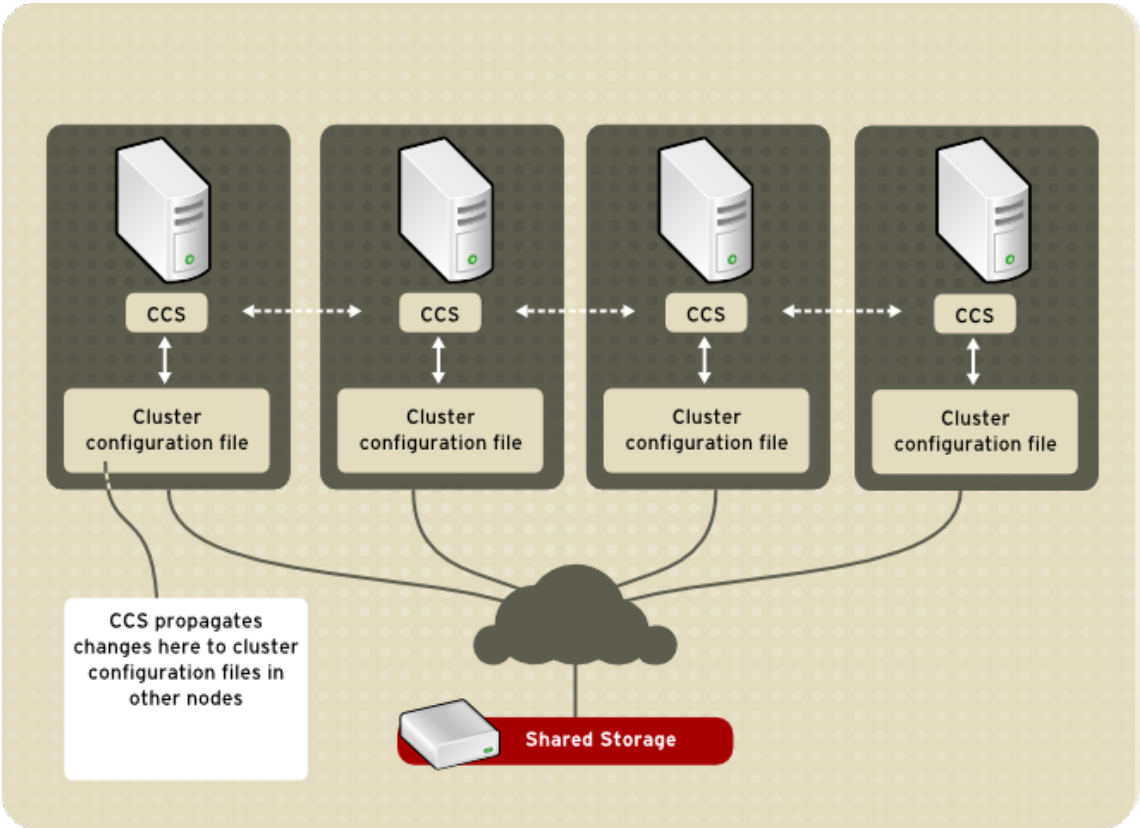


Figura 1.7. CCS Overview

Other cluster components (for example, CMAN) access configuration information from the configuration file through CCS (refer to Figura 1.7, "CCS Overview").

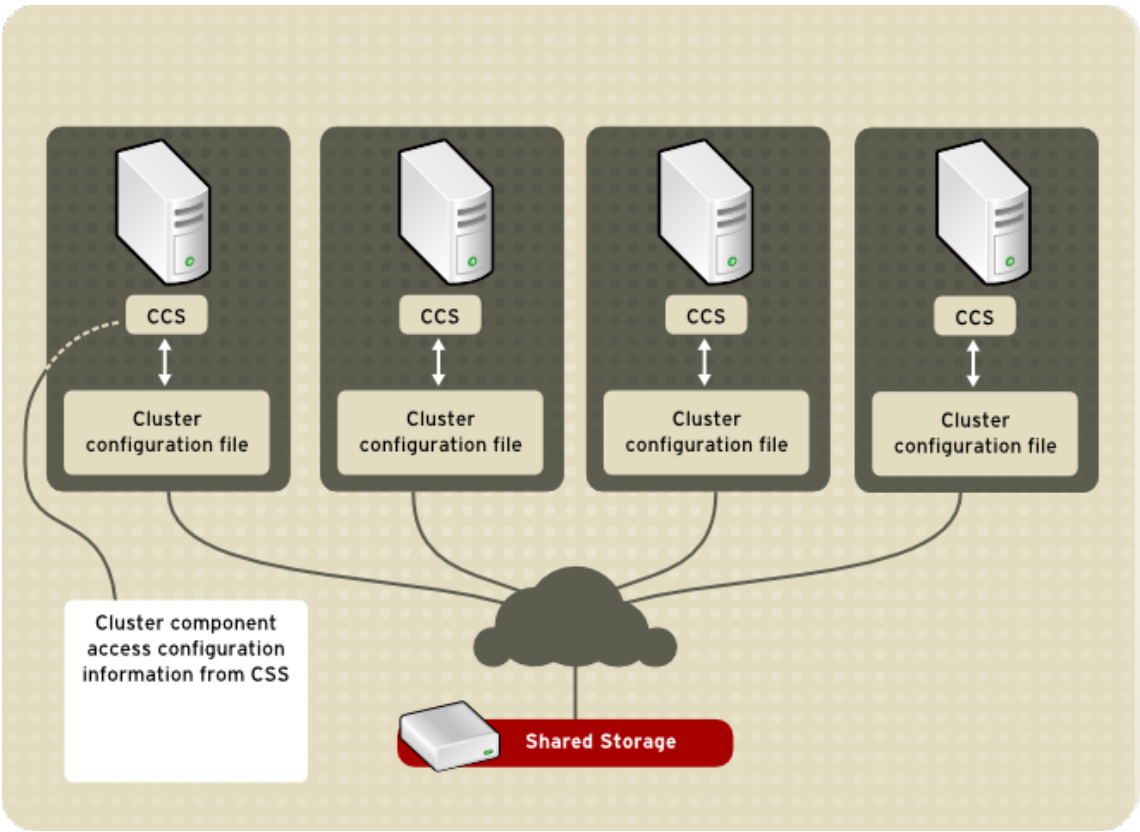


Figura 1.8. Accessing Configuration Information

El archivo de configuración de cluster (`/etc/cluster/cluster.conf`) es un archivo en XML que describe las siguientes características de cluster:

- Nombre de cluster — Muestra el nombre de cluster, el nivel de revisión del archivo de configuración de cluster y las propiedades de aislamiento básico que son utilizadas cuando un nodo entra al cluster o es aislado por el cluster.
- Cluster — Muestra cada nodo de cluster, especifica el nombre del nodo, el ID del nodo, el número de votos en el quórum y el método de aislamiento para el nodo.
- El dispositivo de aislamiento — Muestra el dispositivo de aislamiento en el cluster. Los parámetros pueden variar de acuerdo con el tipo de dispositivo de aislamiento. Por ejemplo, para un controlador de energía que es utilizado como dispositivo de aislamiento, la configuración de cluster define el nombre del controlador de poder, su dirección IP, el login y la contraseña.
- Recursos administrados — Muestra los recursos requeridos para crear los servicios de cluster. Entre los recursos administrados se encuentra la definición del dominio de recuperación contra fallas, los recursos (dirección IP por ejemplo) y los servicios. Los recursos administrados definen los servicios de cluster y el comportamiento de recuperación contra fallas de éstos.

1.4. ADMINISTRACIÓN DE SERVICIOS DE ALTA DISPONIBILIDAD

La administración de servicios de alta disponibilidad proporcionan la habilidad de crear y administrar *servicios de cluster*de alta disponibilidad en un cluster de Red Hat. El componente clave de la administración de servicios de alta disponibilidad en un cluster de Red Hat es **`rgmanager`**. Este componente implementa recuperación contra fallos para aplicaciones. En un cluster de Red Hat, una aplicación es configurada con otros recursos del cluster para formar un servicio de cluster de alta disponibilidad. Un servicio de cluster de alta disponibilidad puede pasar de un nodo a otro sin ninguna interrupción aparente a los clientes de cluster. La recuperación contra fallos puede ocurrir si un nodo de cluster falla o si el administrador de sistema de cluster traslada el servicio de un nodo a otro (por ejemplo si el nodo necesita recibir tareas de mantenimiento).

Para crear un servicio de alta disponibilidad se debe configurar éste en el archivo de configuración de cluster. Un servicio de cluster consta de *recursos* de cluster. Los recursos de cluster son bloques de construcción que se crean y administran en el archivo de configuración de cluster — por ejemplo, una dirección IP, el script de inicialización de una aplicación o una partición compartida Red Hat GFS.

You can associate a cluster service with a *failover domain*. A failover domain is a subset of cluster nodes that are eligible to run a particular cluster service (refer to Figura 1.9, “Dominios de recuperación contra fallos”).

NOTA

Los dominios de recuperación contra fallos *no* son requeridos para el funcionamiento del cluster.

Un servicio de cluster puede ser ejecutado en sólo un nodo de cluster en un momento dado para mantener la integridad de los datos. Se puede especificar la prioridad en el dominio de recuperación contra fallos, asignando niveles de prioridad a cada nodo en el dominio. El nivel de prioridad determina el orden de recuperación contra fallos — en otras palabras determina el nodo que debe reemplazar al nodo fallido. Si no se especifica la prioridad de recuperación contra fallos, un servicio de cluster puede pasar a cualquier nodo dentro del dominio. Asimismo, se puede especificar si un servicio de cluster

debe ser ejecutado únicamente en los nodos de su dominio de recuperación contra fallos asociado. Cuando el servicio está asociado a un dominio de recuperación contra fallos no restringido, un servicio de cluster puede ser iniciado en cualquier nodo si no hay ningún nodo del dominio disponible.

In Figura 1.9, “Dominios de recuperación contra fallos” , Failover Domain 1 is configured to restrict failover within that domain; therefore, Cluster Service X can only fail over between Node A and Node B. Failover Domain 2 is also configured to restrict failover with its domain; additionally, it is configured for failover priority. Failover Domain 2 priority is configured with Node C as priority 1, Node B as priority 2, and Node D as priority 3. If Node C fails, Cluster Service Y fails over to Node B next. If it cannot fail over to Node B, it tries failing over to Node D. Failover Domain 3 is configured with no priority and no restrictions. If the node that Cluster Service Z is running on fails, Cluster Service Z tries failing over to one of the nodes in Failover Domain 3. However, if none of those nodes is available, Cluster Service Z can fail over to any node in the cluster.

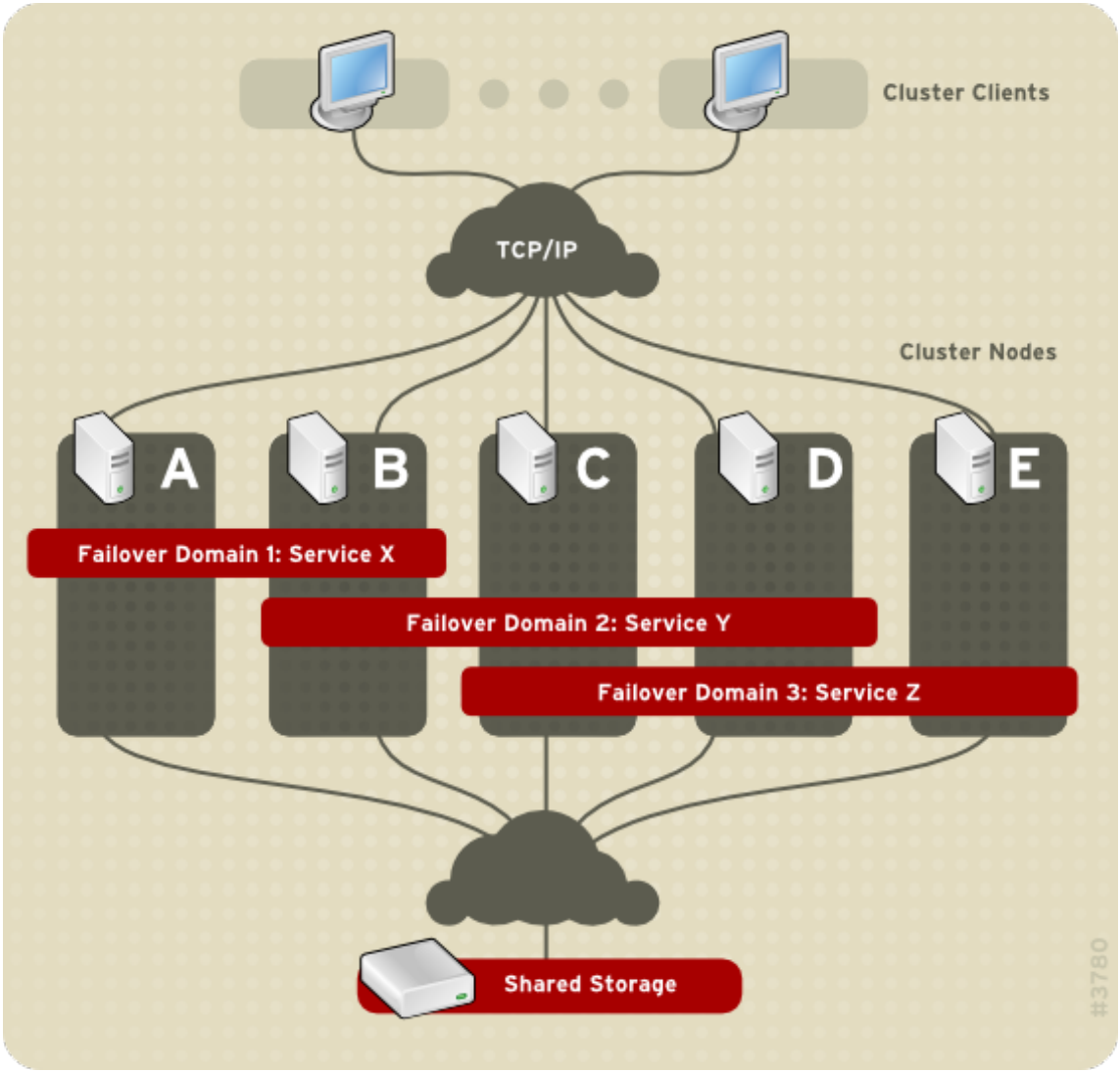


Figura 1.9. Dominios de recuperación contra fallos

Figura 1.10, “Web Server Cluster Service Example” shows an example of a high-availability cluster service that is a web server named "content-webserver". It is running in cluster node B and is in a failover domain that consists of nodes A, B, and D. In addition, the failover domain is configured with a failover priority to fail over to node D before node A and to restrict failover to nodes only in that failover domain. The cluster service comprises these cluster resources:

- recurso de dirección IP — dirección IP 10.10.10.201.

- An application resource named "httpd-content" — a web server application init script `/etc/init.d/httpd` (specifying `httpd`).
- A file system resource — Red Hat GFS named "gfs-content-webserver".

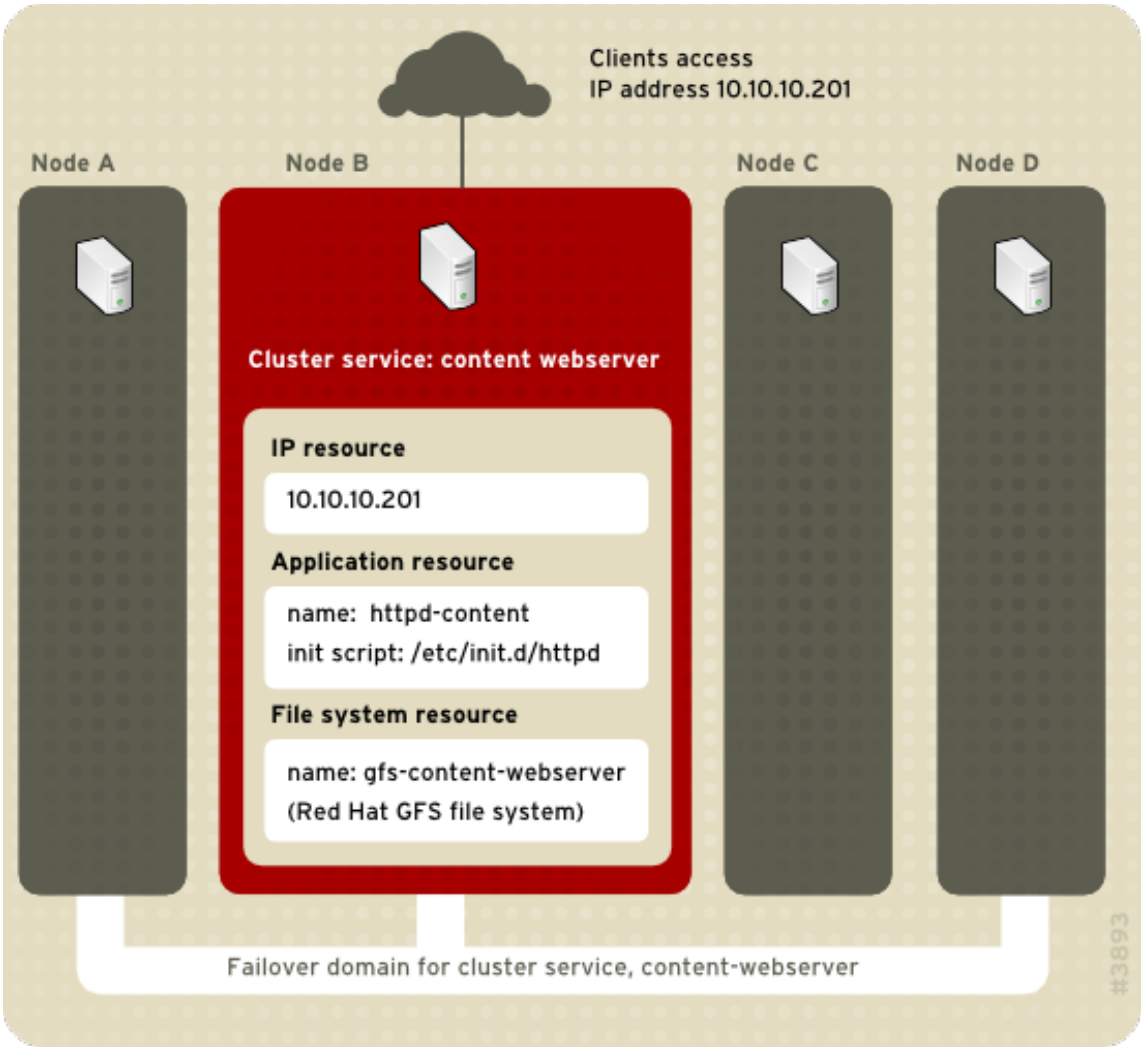


Figura 1.10. Web Server Cluster Service Example

Los clientes acceden al servicio de cluster a través de la dirección IP 10.10.10.201, permitiendo la interacción con la aplicación de servidor web, httpd-content. La aplicación httpd-content utiliza el sistema de archivos gfs-content-webserver. Si el nodo B falla, el servicio de cluster content-webserver pasa al nodo D. Si el nodo D no está disponible o falla, el servicio pasa al nodo A. La recuperación contra fallos no será perceptible por los clientes. Se podrá acceder al servicio de cluster desde otro nodo del cluster a través de la misma dirección IP que se utilizaba antes de la falla.

1.5. RED HAT GFS

Red Hat GFS es un sistema de archivo de cluster que permite que los nodos de un cluster tengan acceso simultáneo a un dispositivo de bloque compartido. GFS es un sistema de archivos nativo que interactúa directamente con la capa VFS de la interfaz del sistema de archivos del kernel de Linux. GFS utiliza metadatos distribuidos y varios diarios de registro (journals) para asegurar la óptima operación en un cluster. Para mantener la integridad del sistema de archivos, GFS utiliza un cierre de exclusión mutua para coordinar las operaciones de E/S. Cuando un nodo cambia los datos en el sistema de archivos GFS, estos son inmediatamente visibles desde los otros nodos del cluster que utilizan el sistema de archivos.

Con Red Hat GFS se puede obtener el mayor tiempo de funcionamiento de una aplicación a través de los siguientes beneficios:

- Simplifica la infraestructura de sus datos
  - Permite la instalación de aplicaciones para todo el cluster.
  - Elimina la necesidad de copias innecesarias de los datos de la aplicación (duplicación)
  - Permite acceso de lectura y escritura concurrente a los datos de varios clientes.
  - Simplifica la creación de copias de seguridad y la recuperación contra desastres (sólo un sistema de archivos debe ser copiado o recuperado).
- Maximiza el uso de recursos de almacenaje; minimiza los costos de administración de almacenaje.
  - Administra el almacenaje como un todo y no como particiones.
  - Decrece el almacenamiento general al eliminar la necesidad de duplicación de datos.
- Escala el cluster al añadir servidores o almacenaje en la marcha.
  - Evita el particionamiento de almacenaje a través de técnicas complicadas.
  - Añade servidores al cluster montándolos en el sistema de archivos común.

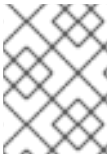
Los nodos que ejecutan Red Hat GFS son configurados y administrados con las herramientas de configuración y administración de Red Hat Cluster Suite. La administración de volúmenes se realiza a través de CLVM (Cluster Logical Volume Manager). Red Hat GFS proporciona compartición de datos entre los nodos GFS en un cluster de Red Hat. GFS proporciona un panorama consistente y único de los espacios de nombre del sistema de archivo a los largo de los nodos GFS en un cluster de Red Hat. GFS permite que las aplicaciones sean instaladas y ejecutadas sin necesidad de un conocimiento detallado de la infraestructura de almacenamiento. Asimismo, GFS proporciona funcionalidades que son típicamente requeridas en entornos empresariales, tales como cuotas, varios diarios de registro y soporte de múltiples rutas.

GFS proporciona un método versátil del almacenamiento de red de acuerdo con el rendimiento, escalabilidad y economía necesarias en su entorno de almacenaje. Este capítulo proporciona información básica y abreviada para ayudar al lector a entender GFS.

You can deploy GFS in a variety of configurations to suit your needs for performance, scalability, and economy. For superior performance and scalability, you can deploy GFS in a cluster that is connected directly to a SAN. For more economical needs, you can deploy GFS in a cluster that is connected to a LAN with servers that use *GNBD* (Global Network Block Device) or to *iSCSI* (Internet Small Computer System Interface) devices. (For more information about GNBD, refer to Sección 1.7, “Dispositivo de bloque de red global (GNBD)”.)

Las siguientes secciones proporcionan ejemplos de cómo GFS puede ser implementado para cubrir las necesidades de rendimiento, escalabilidad y economía:

- Sección 1.5.1, “Rendimiento y escalabilidad superior”
- Sección 1.5.2, “Rendimiento, escalabilidad y precio moderado”
- Sección 1.5.3, “Economía y rendimiento”



NOTA

Los ejemplos de implementaciones GFS reflejan configuraciones básicas; se podría requerir una combinación de éstas para lograr mejores resultados.

1.5.1. Rendimiento y escalabilidad superior

You can obtain the highest shared-file performance when applications access storage directly. The GFS SAN configuration in Figura 1.11, “GFS with a SAN” provides superior file performance for shared files and file systems. Linux applications run directly on cluster nodes using GFS. Without file protocols or storage servers to slow data access, performance is similar to individual Linux servers with directly connected storage; yet, each GFS application node has equal access to all data files. GFS supports over 300 GFS nodes.

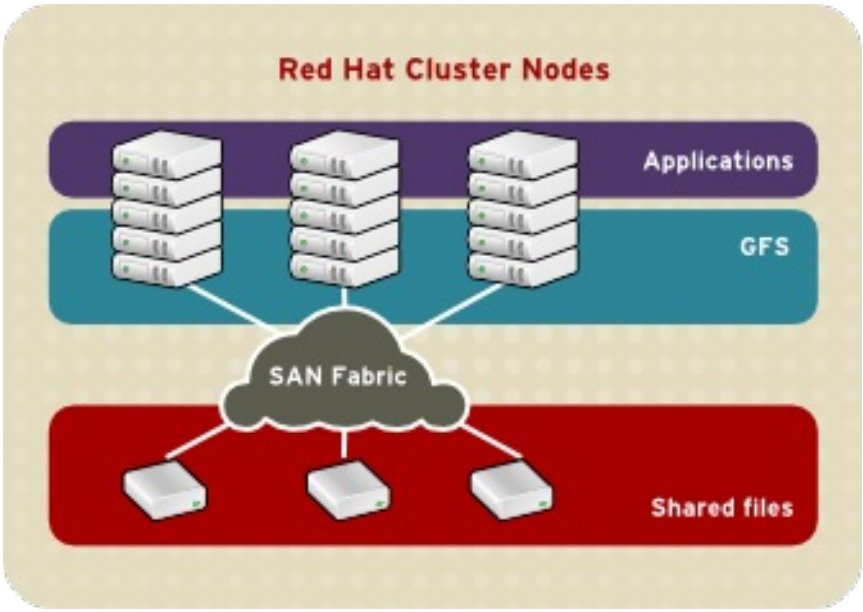


Figura 1.11. GFS with a SAN

1.5.2. Rendimiento, escalabilidad y precio moderado

Multiple Linux client applications on a LAN can share the same SAN-based data as shown in Figura 1.12, “GFS and GNBD with a SAN”. SAN block storage is presented to network clients as block storage devices by GNBD servers. From the perspective of a client application, storage is accessed as if it were directly attached to the server in which the application is running. Stored data is actually on the SAN. Storage devices and data can be equally shared by network client applications. File locking and sharing functions are handled by GFS for each network client.

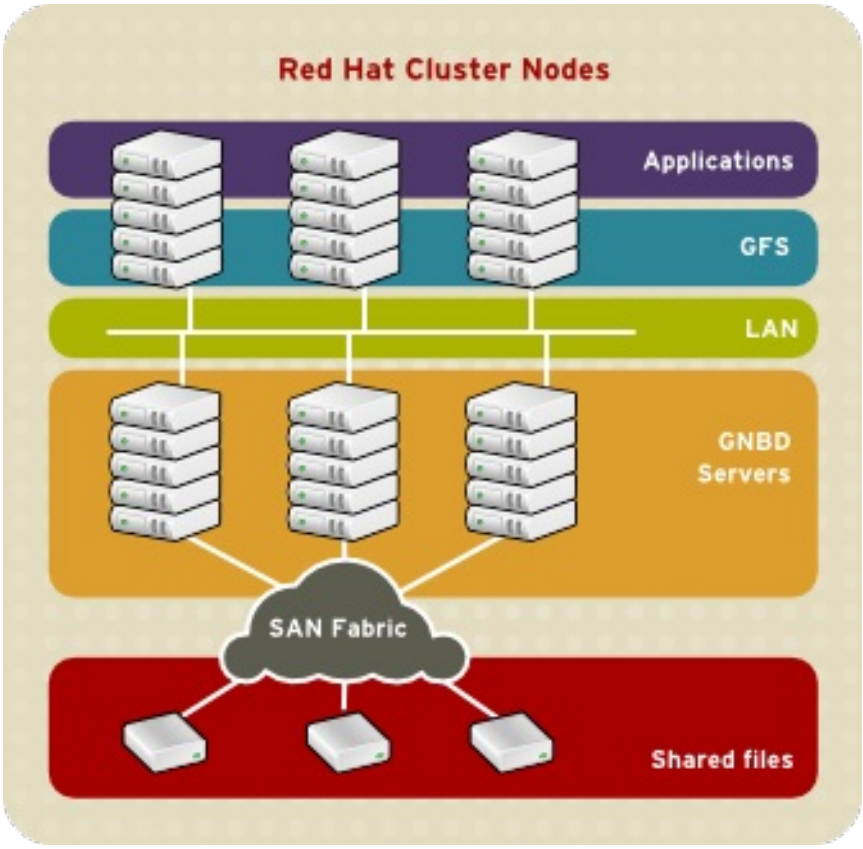


Figura 1.12. GFS and GNBD with a SAN

1.5.3. Economía y rendimiento

Figura 1.13, "GFS y GNDB con un almacenaje conectado directamente" shows how Linux client applications can take advantage of an existing Ethernet topology to gain shared access to all block storage devices. Client data files and file systems can be shared with GFS on each client. Application failover can be fully automated with Red Hat Cluster Suite.

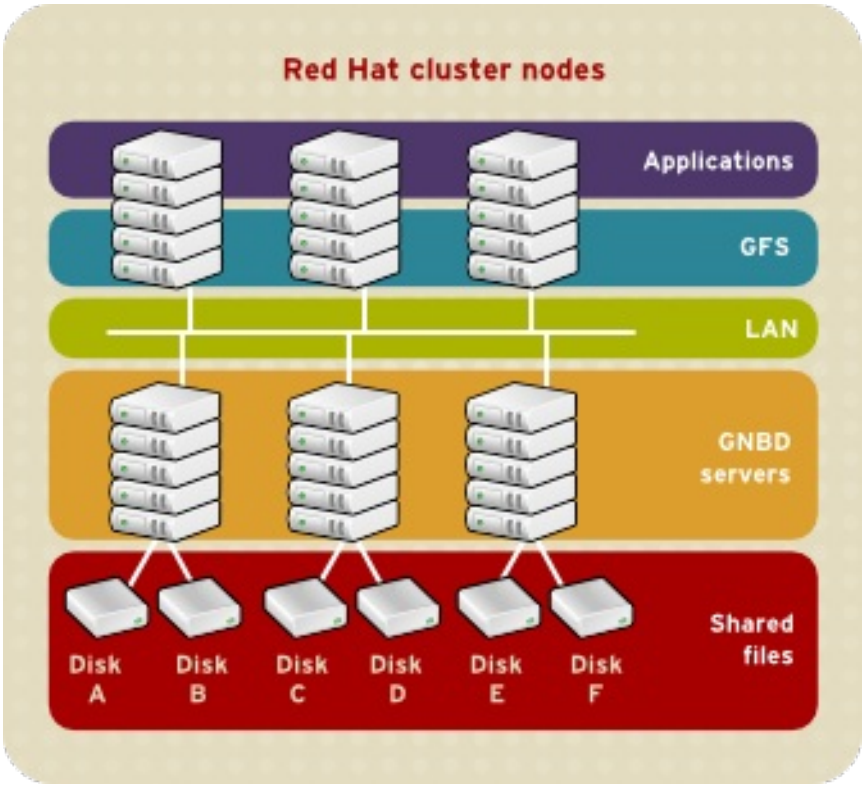
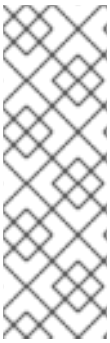


Figura 1.13. GFS y GNDB con un almacenaje conectado directamente

1.6. ADMINISTRADOR DE VOLÚMENES LÓGICOS DE CLUSTER

El administrador de volúmenes lógicos de cluster (CLVM) proporciona una versión de LVM2 a nivel de cluster. CLVM proporciona las mismas funcionalidades que LVM2 en un solo nodo, pero hace que los volúmenes estén disponibles para todos los nodos en un cluster de Red Hat. Los volúmenes lógicos creados con CLVM hacen que los volúmenes lógicos estén disponibles para todos los nodos en un cluster.

The key component in CLVM is **clvmd**. **clvmd** is a daemon that provides clustering extensions to the standard LVM2 tool set and allows LVM2 commands to manage shared storage. **clvmd** runs in each cluster node and distributes LVM metadata updates in a cluster, thereby presenting each cluster node with the same view of the logical volumes (refer to Figura 1.14, “CLVM Overview”). Logical volumes created with CLVM on shared storage are visible to all nodes that have access to the shared storage. CLVM allows a user to configure logical volumes on shared storage by locking access to physical storage while a logical volume is being configured. CLVM uses the lock-management service provided by the cluster infrastructure (refer to Sección 1.3, “Cluster Infrastructure”).



NOTA

El almacenamiento compartido para uso en Red Hat Cluster Suite requiere que usted esté ejecutando el daemon de administrador de volúmenes lógicos de cluster (**clvmd**) o los agentes de administración de volúmenes lógicos de alta disponibilidad (HA-LVM). Si no puede utilizar ni el daemon **clvmd** ni HA-LVM por razones operativas o porque no tiene la debida autorización, no debe utilizar la instancia-única de LVM en el disco compartido porque se pueden dañar los datos. Si tiene dudas, por favor contacte al representante de servicio de Red Hat.

NOTA

El uso de CLVM requiere cambios menores a `/etc/lvm/lvm.conf` para el sistema de cierres de exclusión de cluster.

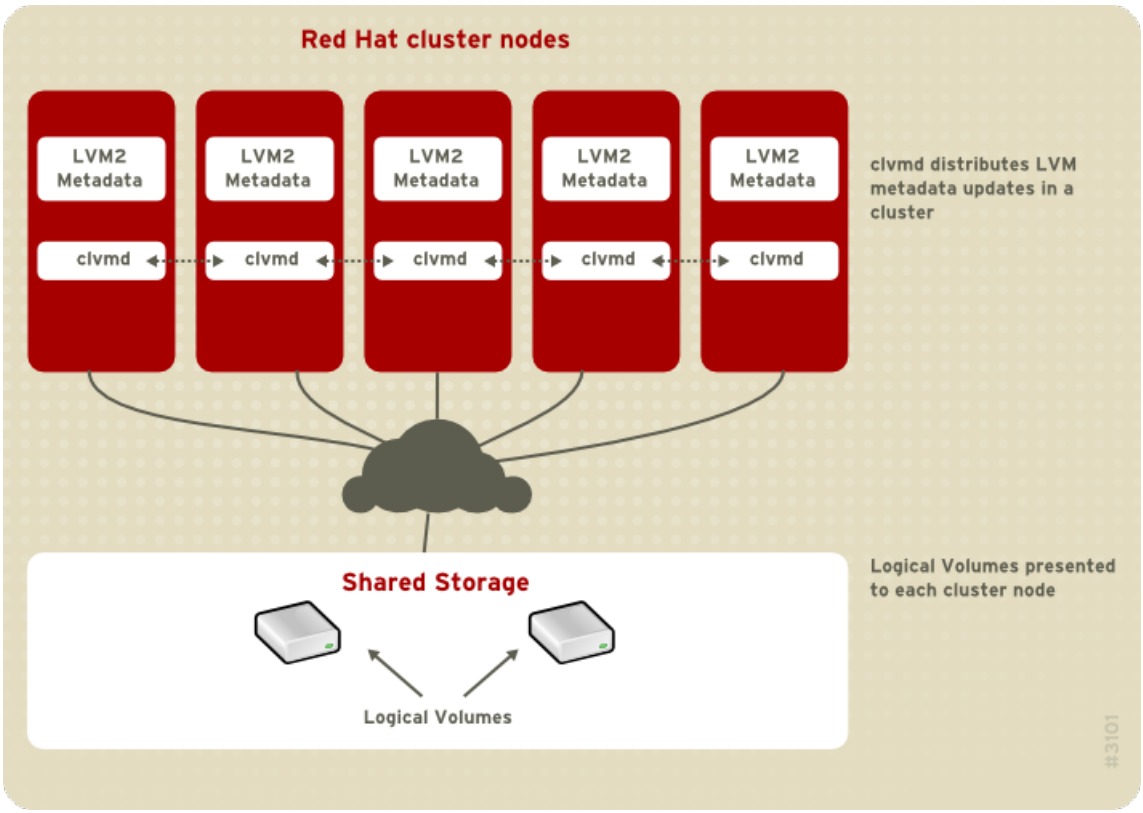


Figura 1.14. CLVM Overview

You can configure CLVM using the same commands as LVM2, using the LVM graphical user interface (refer to Figura 1.15, “LVM Graphical User Interface”), or using the storage configuration function of the **Conga** cluster configuration graphical user interface (refer to Figura 1.16, “Conga LVM Graphical User Interface”) . Figura 1.17, “Creating Logical Volumes” shows the basic concept of creating logical volumes from Linux partitions and shows the commands used to create logical volumes.

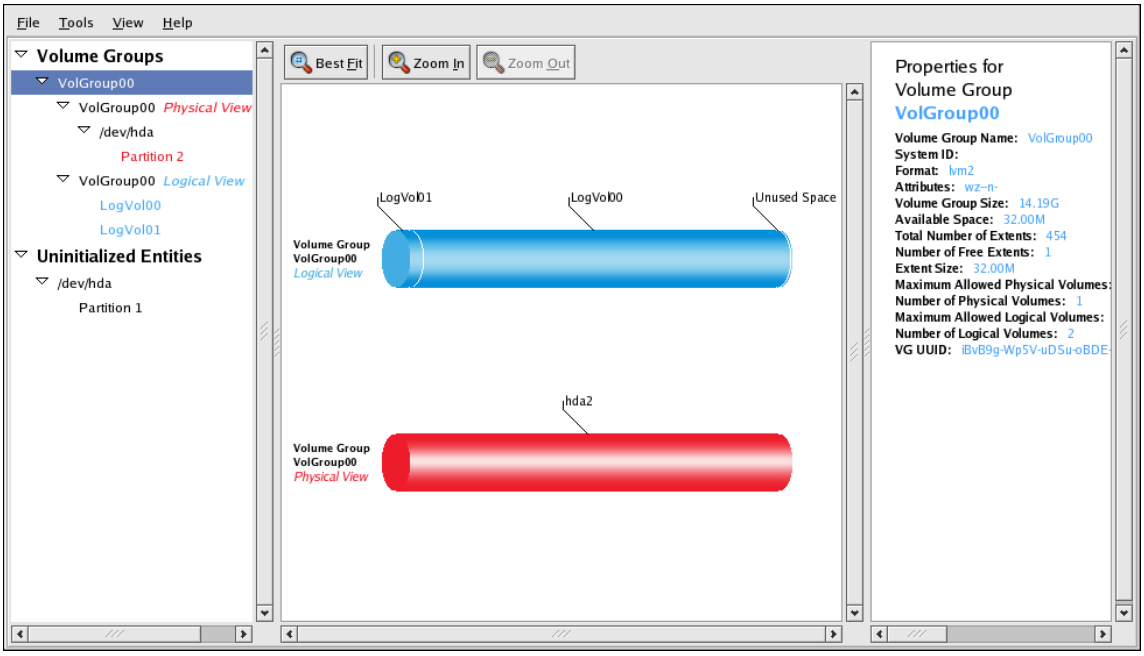


Figura 1.15. LVM Graphical User Interface

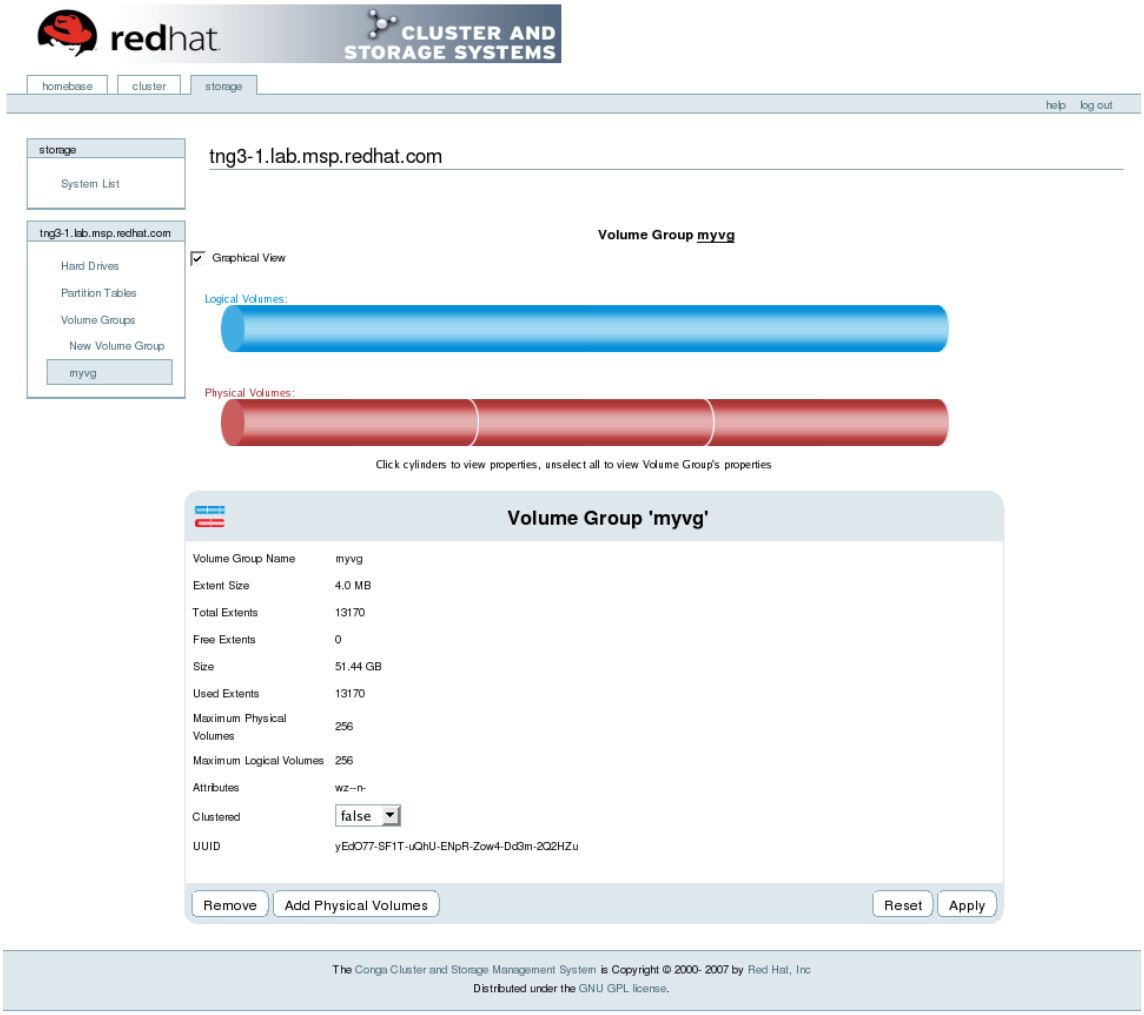


Figura 1.16. Conga LVM Graphical User Interface

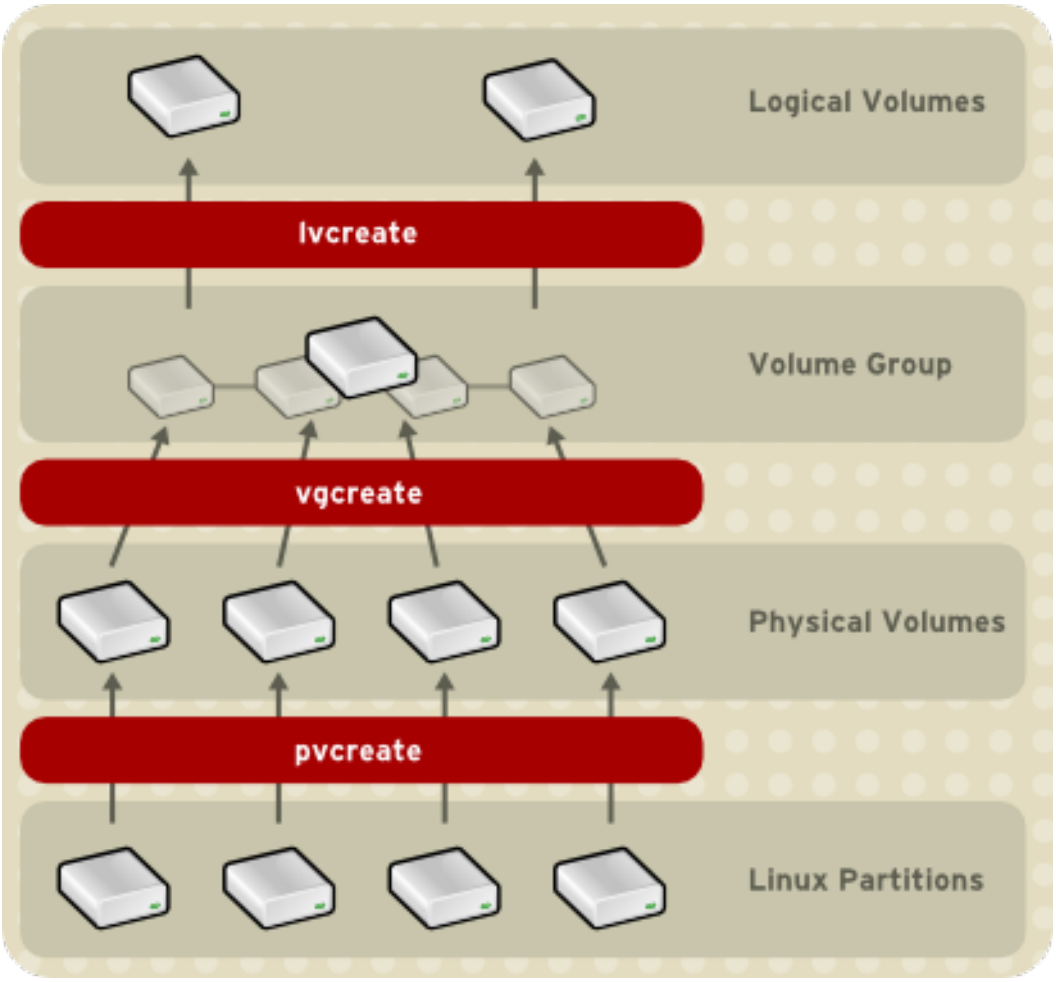


Figura 1.17. Creating Logical Volumes

1.7. DISPOSITIVO DE BLOQUE DE RED GLOBAL (GNBD)

EL dispositivo de bloque de red global (GNDB) proporciona acceso de dispositivo de bloque para Red Hat GFS sobre TCP/IP. GNBD es un concepto similar a NBD; sin embargo, GNBD es específico de GFS y está diseñado para ser utilizado específicamente con GFS. GNBD es útil cuando la necesidad de tecnologías más robustas — canal de fibra o iniciadores SCSI sencillos — no son necesarias o no viables económicamente.

GNBD consists of two major components: a GNBD client and a GNBD server. A GNBD client runs in a node with GFS and imports a block device exported by a GNBD server. A GNBD server runs in another node and exports block-level storage from its local storage (either directly attached storage or SAN storage). Refer to Figura 1.18, “Sinopsis de GNBD”. Multiple GNBD clients can access a device exported by a GNBD server, thus making a GNBD suitable for use by a group of nodes running GFS.

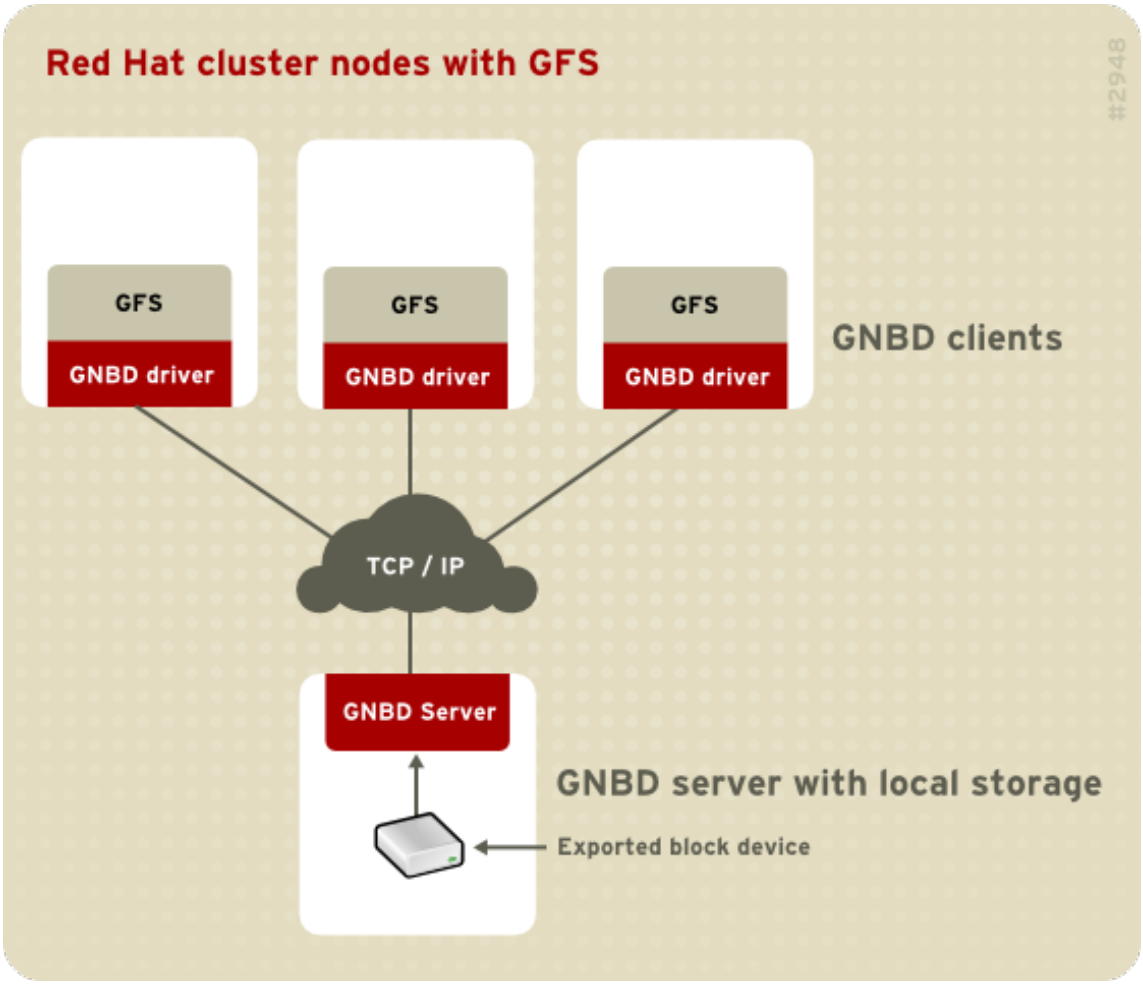


Figura 1.18. Sinopsis de GNBD

1.8. SERVIDOR VIRTUAL DE LINUX

El servidor virtual de Linux (LVS) es un grupo de componentes de software integrado para balancear la carga de IP a lo largo de un conjunto de servidores reales. LVS se ejecuta en un par de computadores configurados de la misma forma: uno que funciona como un enrutador LVS activo y otro que funciona como un enrutador LVS de respaldo. El enrutador LVS activo tiene dos roles:

- Balancear la carga entre los servidores reales.
- Revisar la integridad de los servicios en cada servidor real.

El enrutador LVS de respaldo sondea el estado del enrutador LVS activo y toma el control de sus tareas en caso de que éste falle.

Figura 1.19, “Components of a Running LVS Cluster” provides an overview of the LVS components and their interrelationship.

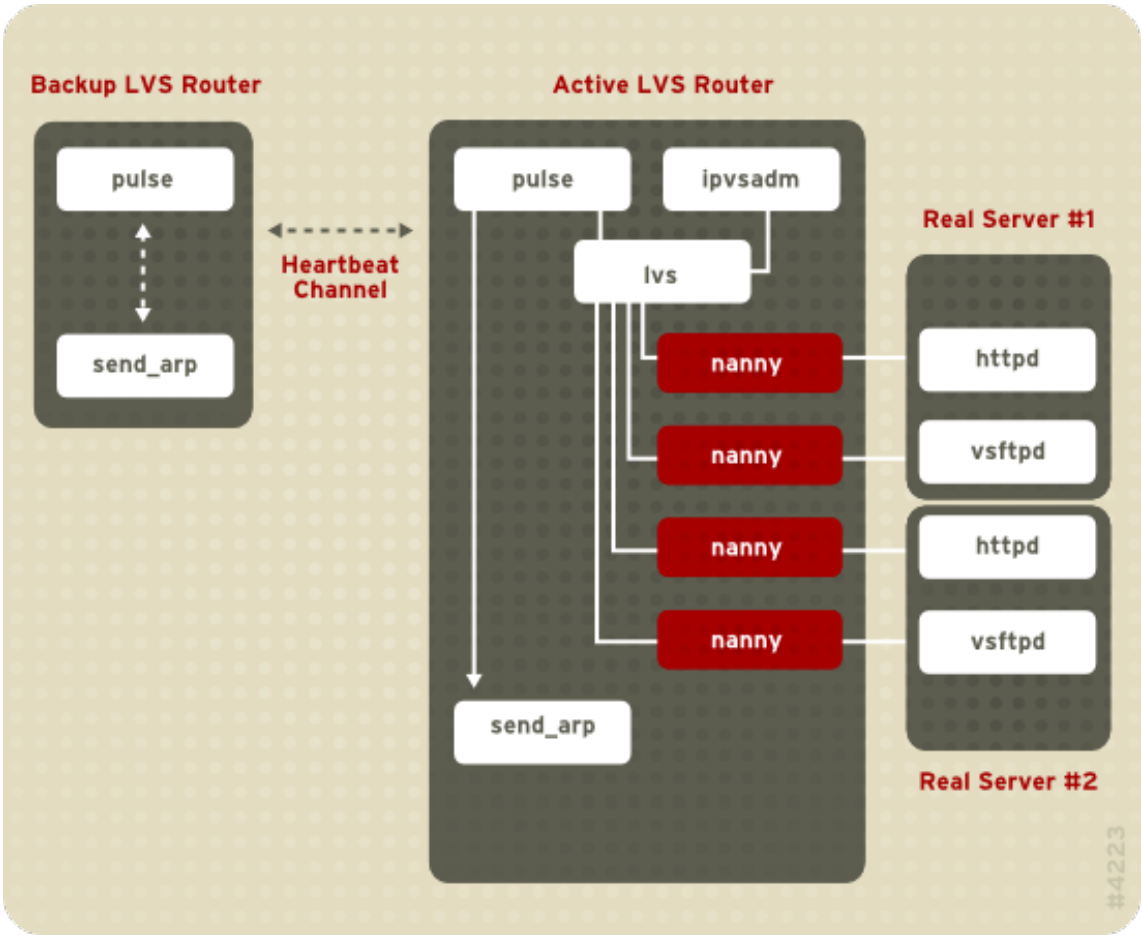


Figura 1.19. Components of a Running LVS Cluster

El daemon **pulse** se ejecuta tanto en el servidor LVS activo como en el pasivo. En el enrutador LVS de respaldo, **pulse** envía un *latido* a la interfaz pública del enrutador LVS activo para asegurarse de que éste esté funcionando apropiadamente. En el enrutador LVS activo, **pulse** inicia el daemon **lvs** y responde a los *latidos* que provienen del enrutador LVS de respaldo.

Una vez iniciado, el daemon **lvs** llama a la utilidad **ipvsadmin** para configurar y mantener la tabla de rutas IPVS (IP Virtual Server) en el kernel e inicia un proceso **nanny** para cada servidor virtual configurado en cada servidor real. Cada proceso **nanny** revisa el estado de cada servidor configurado en un servidor real e informa al daemon **lvs** si el servicio en el servidor real no está funcionando. Si el servicio no está funcionando, el daemon **lvs** ordena a **ipvsadm** que remueva el servidor real de la tabla de rutas IPVS.

Si el enrutador LVS de respaldo no recibe una respuesta desde el enrutador LVS activo, el primero inicia un proceso de recuperación contra fallos llamando a **send_arp** para que asigne nuevamente todas las direcciones IP virtuales a las direcciones de hardware NIC (direcciones *MAC*) del enrutador LVS de respaldo, envía un comando para activar el enrutador LVS activo a través de las interfaces de red pública y privada para apagar el daemon **lvs** en el enrutador LVS activo e inicia el daemon **lvs** en el enrutador LVS de respaldo para que acepte solicitudes para los servidores virtuales configurados.

Para un usuario externo que accede al servicio hospedado (tal como un sitio web o una base de datos), LVS aparece como un servidor. Sin embargo, el usuario está accediendo al servidor real tras los enrutadores LVS.

Ya que no existen componentes internos en LVS para compartir los datos entre servidores reales, hay dos opciones básicas:

- Sincronizar los datos entre los servidores reales.

- Añadir una tercera capa a la topología para el acceso de datos compartidos.

La primera opción es la preferida en aquellos servidores que no permiten a un gran número de usuarios cargar o cambiar datos en el servidor real. Si los servidores reales permiten que los datos sean modificados por un gran número de usuarios, por ejemplo los sitios web de comercio electrónico, es preferible añadir una nueva capa.

Hay varios métodos para sincronizar los datos entre los servidores reales. Por ejemplo, puede utilizar un script de shell para publicar las páginas web actualizadas a los servidores reales de forma simultánea. Asimismo, puede utilizar programas como **rsync** para replicar los cambios de datos a lo largo de todos los nodos cada cierto intervalo de tiempo. Sin embargo, en los entornos donde los usuarios cargan archivos o ejecutan transacciones a la base de datos, el uso de scripts o del comando **rsync** para la sincronización de datos no funciona de forma óptima. Por lo cual, para servidores reales con una gran cantidad de cargas, transacciones a bases de datos o tráfico similar, una *topología de tres capas* es la opción más apropiada para la sincronización de datos.

1.8.1. Two-Tier LVS Topology

Figura 1.20, “Two-Tier LVS Topology” shows a simple LVS configuration consisting of two tiers: LVS routers and real servers. The LVS-router tier consists of one active LVS router and one backup LVS router. The real-server tier consists of real servers connected to the private network. Each LVS router has two network interfaces: one connected to a public network (Internet) and one connected to a private network. A network interface connected to each network allows the LVS routers to regulate traffic between clients on the public network and the real servers on the private network. In Figura 1.20, “Two-Tier LVS Topology”, the active LVS router uses *Network Address Translation* (*NAT*) to direct traffic from the public network to real servers on the private network, which in turn provide services as requested. The real servers pass all public traffic through the active LVS router. From the perspective of clients on the public network, the LVS router appears as one entity.

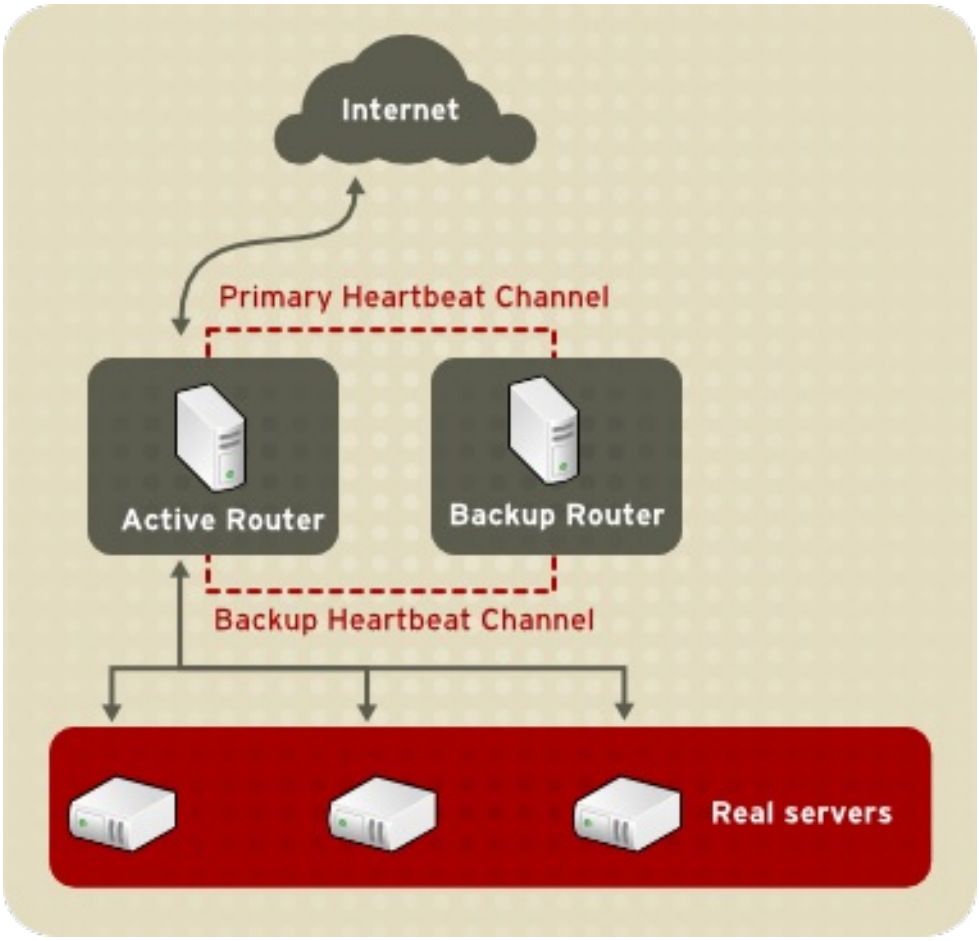


Figura 1.20. Two-Tier LVS Topology

Los servicios solicitados a un enrutador LVS son dirigidos a una dirección *IP virtual* o VIP. Esta es una dirección enrutada públicamente que el administrador del sitio asocia con el nombre de dominio totalmente calificado, tal como www.example.com, y que se asigna a uno o más *servidores virtuales*[1]. Observe que una dirección VIP migra de un enrutador LVS a otro durante el proceso de recuperación contra fallos. Esto hace que siempre haya una presencia en la dirección IP (conocida también como *dirección IP flotante*).

Las direcciones VIP pueden tener sobrenombres que se dirijan al mismo dispositivo que conecta al enrutador LVS con la red pública. Por ejemplo, si eth0 está conectado a Internet, puede haber varios servidores virtuales con sobrenombres a **`eth0:1`**. Alternativamente, cada servidor virtual puede estar asociado con un dispositivo separado por servicio. Por ejemplo, el tráfico HTTP puede ser manejado en **`eth0:1`** y el tráfico FTP puede ser manejado en **`eth0:2`**.

Solo un enrutador LVS está activo a la vez. El rol del enrutador LVS activo es redireccionar la solicitud del servicio desde la dirección IP virtual al servidor real. La redirección está basada en uno de ocho algoritmos de balance de carga:

- Programador Round-Robin — Distribuye cada solicitud secuencialmente alrededor de los servidores reales. Al usar este algoritmo, todos los servidores reales son tratados del mismo modo, sin importar su capacidad o carga.

- Programador Weighted Round-Robin — Distribuye cada solicitud secuencialmente alrededor de los servidores reales dando más tareas a los servidores con mayor capacidad. La capacidad es indicada por el usuario y se ajusta gracias a la información de carga dinámica. Esta es la opción preferida si los servidores reales tienen distintas capacidades. Sin embargo, si la carga de solicitudes cambia dramáticamente, un servidor con gran capacidad podría responder a más solicitudes que las que debe.

- Least-Connection — Distribuye más solicitudes a los servidores reales que tienen menos conexiones activas. Este es un tipo de algoritmo de programación dinámico. Es una buena opción si hay altos grados de variación en las solicitudes. Es ideal en las infraestructuras donde cada servidor tiene aproximadamente la misma capacidad. Si los servidores reales tienen capacidades variadas, la programación weighted least-connection es una mejor opción.

- Weighted Least-Connections (predeterminado) — Distribuye más solicitudes a los servidores con menos conexiones activas en relación con sus capacidades. La capacidad es indicada por el usuario y es ajustada por la información de carga dinámica. La adición del parámetro de capacidad hace que este algoritmo sea ideal cuando la infraestructura tiene servidores reales con capacidades de hardware variado.

- Locality-Based Least-Connection Scheduling — Distribuye más solicitudes a los servidores con menos conexiones activas en relación con sus IP de destino. Este algoritmo se utiliza en cluster de servidores de caché proxy. Enruta el paquete para una dirección IP para el servidor con esa dirección a menos que el servidor esté sobrecargado, en dicho caso se asigna la dirección IP al servidor real con menos carga.

- Locality-Based Least-Connection Scheduling with Replication Scheduling — Distribuye más solicitudes a los servidores con menos conexiones activas de acuerdo con la IP de destino. Este algoritmo es usado en servidores de caché de proxy. Se diferencia de "Locality-Based Least-Connection Scheduling" al relacionar la dirección IP objetivo con un grupo de servidores reales. Las solicitudes son luego enviadas al servidor en el grupo con menos número de conexiones. Si la capacidad de todos los nodos para el IP de destino está sobre el límite, añade un nuevo servidor real del grupo general al grupo de servidores para el IP de destino. El nodo con mayor carga es desplazado fuera del grupo para evitar un exceso de replicación.

- Source Hash Scheduling — Distribuye todas las solicitudes de acuerdo con un diccionario estático de direcciones IP. Este algoritmo se utiliza en enrutadores LVS con varios cortafuegos.

Asimismo, el enrutador LVS activo sondea dinámicamente la salud de los servicios especificados en los servidores reales a través de un *script de envío y espera*. Para ayudar en la detección de servicios que requieren datos dinámicos, tal como HTTPS o SSL, se puede incluso llamar a programas ejecutables externos. Si un servicio en un servidor real no funciona adecuadamente, el enrutador LVS activo no envía solicitudes a ese servidor hasta que retorne a la operación normal.

El enrutador LVS de respaldo cumple el rol de asistente del sistema. Periódicamente, el enrutador LVS intercambia mensajes llamados pulsos a través de la interfaz pública externa primaria y, en caso de procesos de recuperación contra fallos, a través de la interfaz privada. Si el enrutador LVS de respaldo no recibe un pulso dentro de un intervalo de tiempo determinado, éste inicia el proceso de recuperación contra fallos y asume el rol del enrutador LVS activo. Durante el proceso de recuperación, el enrutador LVS de respaldo toma la dirección VIP servida por el enrutador fallido utilizando una técnica llamada *suplantación de identidad ARP*— en donde el enrutador LVS de respaldo se anuncia como el servidor de destino para los paquetes IP dirigidos al nodo fallido. Cuando el nodo fallido retorna al servicio activo, el enrutador LVS de respaldo asume su rol de asistente de nuevo.

The simple, two-tier configuration in Figura 1.20, “Two-Tier LVS Topology” is suited best for clusters serving data that does not change very frequently — such as static web pages — because the individual real servers do not automatically synchronize data among themselves.

1.8.2. Three-Tier LVS Topology

Figura 1.21, “Three-Tier LVS Topology” shows a typical three-tier LVS configuration. In the example, the active LVS router routes the requests from the public network (Internet) to the second tier — real servers. Each real server then accesses a shared data source of a Red Hat cluster in the third tier over

the private network.

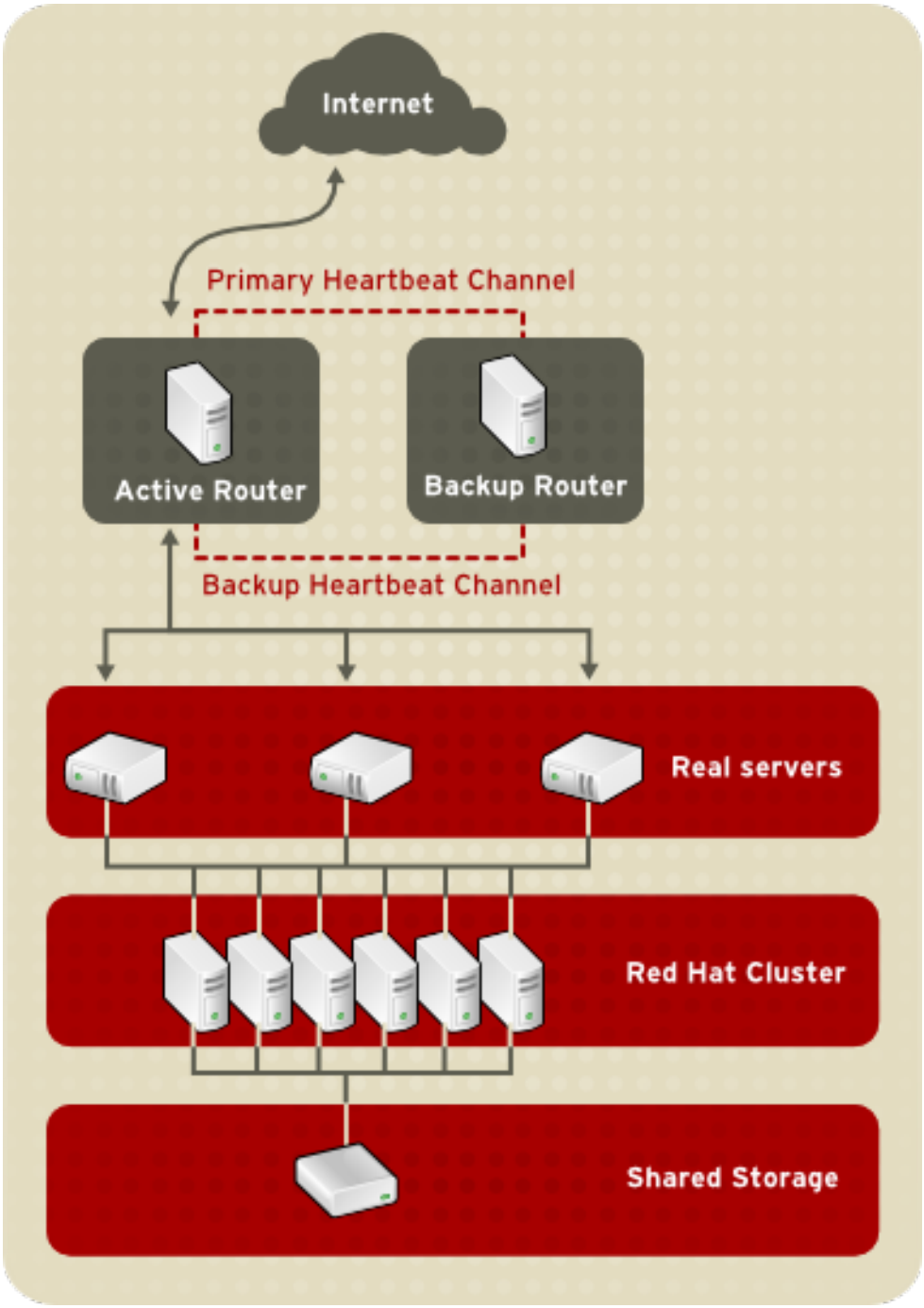


Figura 1.21. Three-Tier LVS Topology

Esta topología es ideal para servidores FTP bastante activos, en donde los datos son almacenados en un servidor central de alta disponibilidad y pueden ser accedidos por cada servidor real a través de un directorio Samba o NFS compartido. Esta topología también es recomendada para sitios web que acceden a una base de datos central de alta disponibilidad para realizar transacciones. Además, al utilizar una configuración activo-activo con un cluster de Red Hat, se puede configurar un cluster de alta disponibilidad al servir los dos roles al mismo tiempo.

1.8.3. Métodos de enrutado

Puede utilizar enrutado NAT (Network Address Translation) o enrutado directo con LVS. La siguiente sección describe abreviadamente ambos enrutados:

1.8.3.1. Enrutado NAT

Figura 1.22, “LVS Implemented with NAT Routing” , illustrates LVS using NAT routing to move requests between the Internet and a private network.

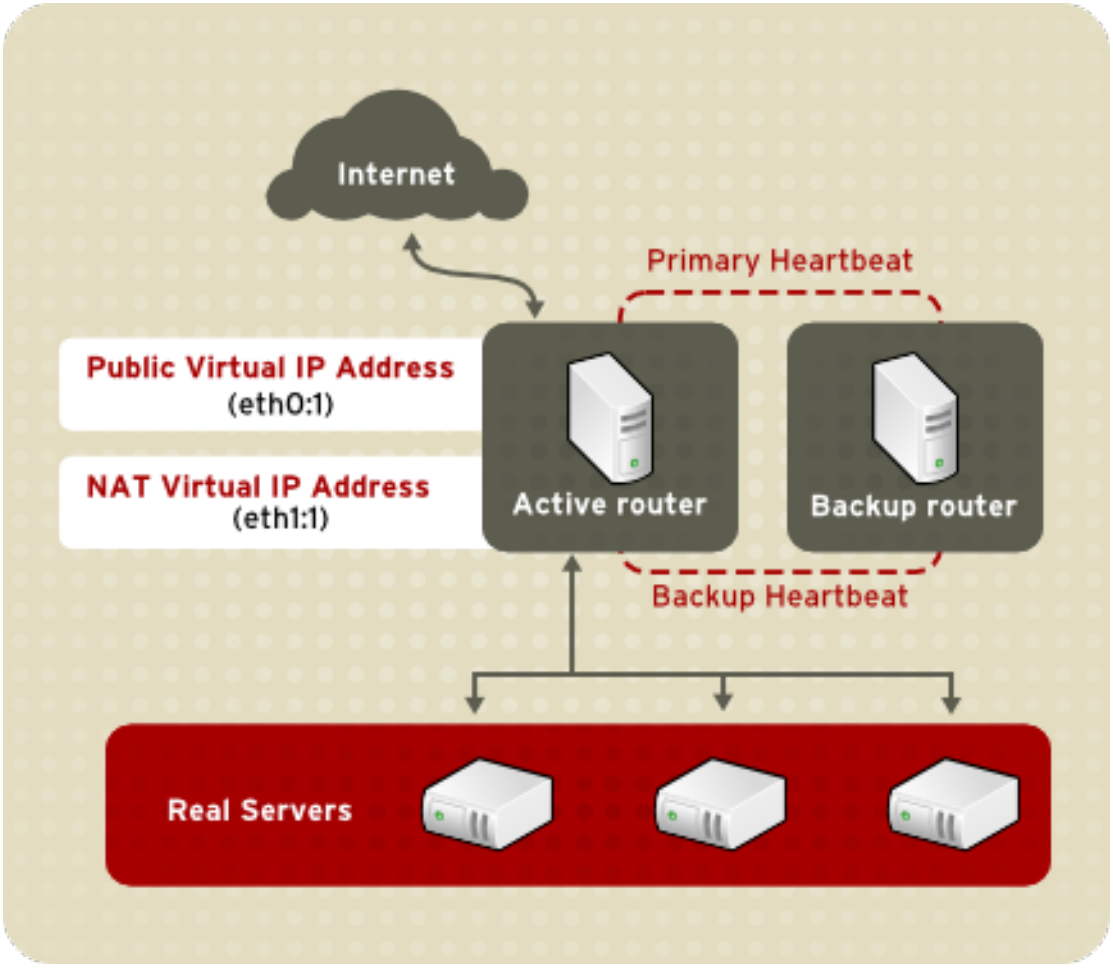


Figura 1.22. LVS Implemented with NAT Routing

En el ejemplo, hay dos NIC en el enrutador LVS activo. El NIC para Internet tiene una *dirección IP real* en eth0 y tiene una dirección IP flotante en eth0:1. El NIC para la interfaz de red privada tiene una dirección IP real en eth1 y tiene una dirección flotante en eth1:1. En el caso de fallo, la interfaz virtual que encara el internet y la privada que encara la interfaz virtual son tomadas simultáneamente por el enrutador LVS de respaldo. Todos los servidores reales en la red privada utilizan la IP flotante para el enrutador NAT como su enrutador predeterminado para comunicarse con el enrutador LVS activo, de esta forma la habilidad para responder a solicitudes desde Internet no se ve impedida.

In the 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 physical device on the LVS router nodes, having more than two NICs is not a requirement.

Con esta topología, el enrutador LVS activo recibe la solicitud y la enruta al servidor apropiado. El servidor real procesa la solicitud y retorna el paquete para el enrutador LVS. El enrutador LVS utiliza nat para remplazar la dirección del servidor real en los paquetes con la dirección VIP pública del enrutador LVS. Este proceso se llama *enmascaramiento de IP*porque la dirección IP de los servidores reales se esconde de los clientes

Al utilizar NAT, los servidores reales pueden ser cualquier computador ejecutando cualquier sistema operativo. La mayor desventaja de NAT es que el enrutador LVS puede volverse un cuello de botella en implementaciones grandes porque éste debe procesar solicitudes entrantes y salientes.

1.8.3.2. Enrutado directo

El enrutado directo proporciona un mejor rendimiento comparado con NAT. El enrutado directo permite que los servidores reales procesen y enruten los paquetes directamente al usuario que los solicitó en vez de pasar los paquetes salientes al enrutador LVS. El enrutado directo reduce la posibilidad de problemas de rendimiento de red ya que el enrutador LVS sólo procesa los paquetes entrantes.

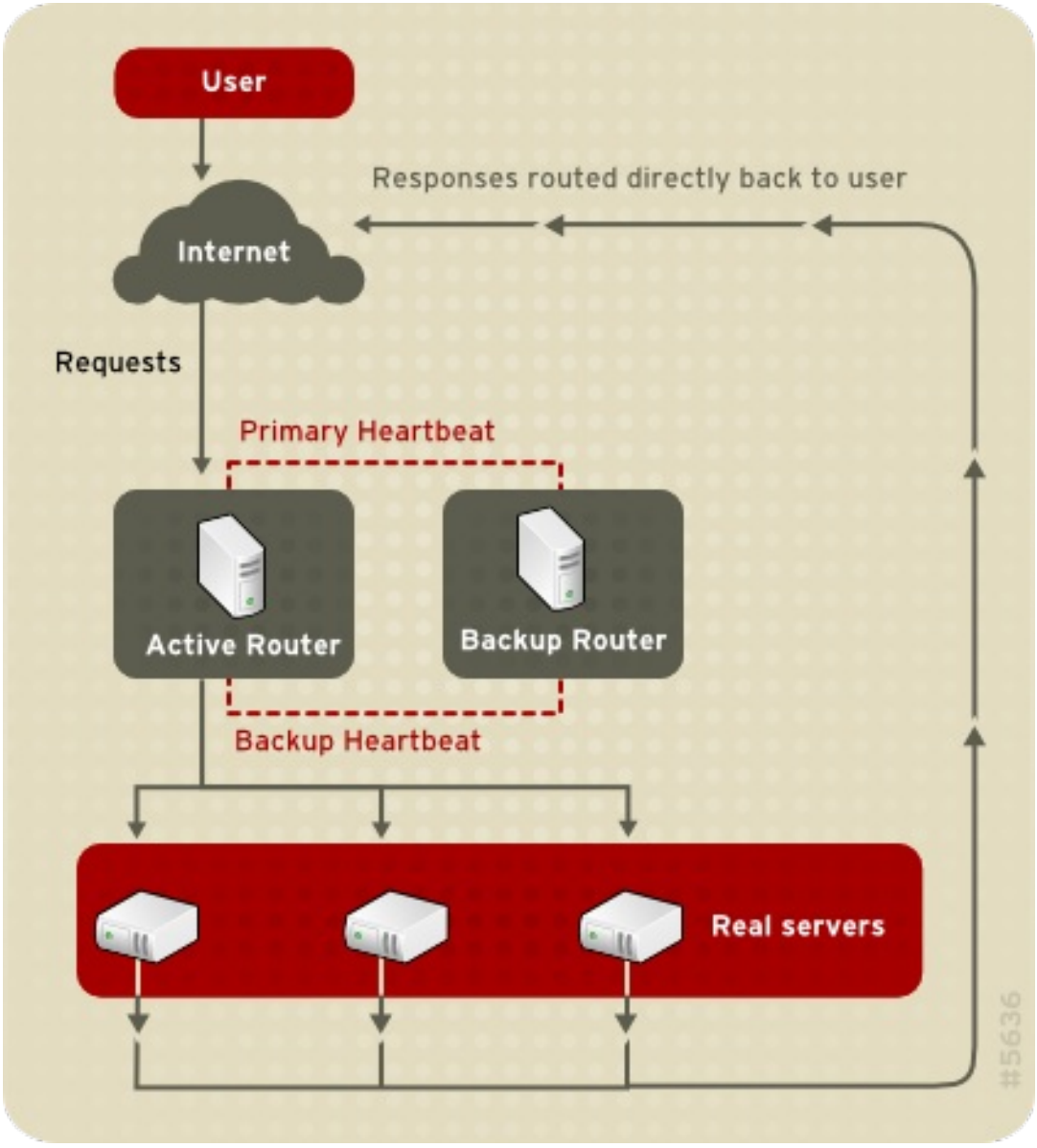


Figura 1.23. LVS Implemented with Direct Routing

En una configuración LVS de enrutado directo, un enrutador LVS recibe una solicitud entrante a través de una IP virtual (VIP) y utiliza un algoritmo para enrutar la solicitud a los servidores reales. Cada servidor real procesa las solicitudes y envía las respuestas directamente a los clientes, sin regresar al enrutador LVS. El enrutamiento directo permite mayor escalabilidad ya que se pueden añadir servidores reales sin que estos tengan que pasar los paquetes salientes al enrutador LVS antes de que lleguen al cliente.

Aunque hay muchas ventajas en utilizar enrutamiento directo en LVS, hay también algunas limitaciones. El problema más común es ARP (siglas en inglés de *Address Resolution Protocol*)

In typical situations, a client on the Internet sends a request to an IP address. Network routers typically send requests to their destination by relating IP addresses to a machine's MAC address with ARP. ARP

requests are broadcast to all connected machines on a network, and the machine with the correct IP/MAC address combination receives the packet. The IP/MAC associations are stored in an ARP cache, which is cleared periodically (usually every 15 minutes) and refilled with IP/MAC associations.

El problema con las solicitudes de ARP en una configuración LVS de enrutado directo es que como una solicitud de cliente a una dirección IP debe estar asociada con una dirección MAC para que la solicitud sea procesada, la dirección IP virtual del enrutador LVS debe estar asimismo asociada con un MAC. Sin embargo, porque tanto el enrutador LVS y los servidores reales tienen el mismo VIP, la solicitud ARP es enviada a todos los nodos asociados con el VIP. Esto puede causar varios problemas, el VIP puede estar asociado directamente con uno de los servidores reales y puede procesar la solicitud directamente, dejando completamente de lado el enrutador LVS y frustrando así la configuración LVS. Al tener un enrutador LVS con una CPU poderosa que pueda responder rápidamente a las solicitudes de los clientes no soluciona necesariamente el problema. Si el enrutador LVS está bajo cargas pesadas, éste puede responder a las solicitudes ARP más lentamente que un servidor real sin mucha carga, el cual responde más rápidamente y se le asigna el VIP en el caché ARP de la solicitud del cliente.

Para solucionar este problema, las solicitudes entrantes deben asociar *únicamente* el VIP y el servidor LVS. Este último procesará adecuadamente las solicitudes y las enviará a los servidores reales. Esto puede realizarse a través de las herramientas de filtrado de paquetes **`arptables`**.

1.8.4. Marcas de cortafuego y persistencia

En algunas circunstancias, puede desearse que un cliente se conecte con el mismo servidor real varias veces en vez de tener que pasar a través de los algoritmos de balance de carga de LVS para encontrar el mejor servidor disponible. Ejemplos de tales situaciones incluyen los formularios web de varias páginas, las cookies, las conexiones SSL y FTP. En dichos casos, el cliente puede no funcionar adecuadamente a menos que la transacción sea procesada por el mismo servidor que retiene el contexto inicial. LVS proporciona dos funcionalidades diferentes para manejar estos casos: *persistencia* y *marcas de cortafuego*.

1.8.4.1. Persistence

Cuando se activa, la persistencia actúa como un contador. Cuando un cliente se conecta a un servicio, LVS recuerda la última conexión para el periodo de tiempo especificado. Si la misma dirección IP de cliente se conecta dentro del periodo de tiempo establecido, la solicitud se envía al mismo servidor que estaba procesando la solicitud anteriormente — dejando de lado el mecanismo de balance de carga. Cuando ocurre una conexión fuera del tiempo límite, ésta se maneja de acuerdo con las reglas de programación en uso.

La persistencia también permite especificar una máscara de subred para aplicar a las direcciones IP del cliente como herramienta para controlar las direcciones que tienen mayor nivel de persistencia, agrupando así conexiones a esa subred.

Al agrupar conexiones destinadas a diferentes puertos puede ser importante para los protocolos que utilizan más de un puerto para comunicarse, tal como FTP. Sin embargo, la persistencia no es la manera más efectiva de agrupar las conexiones destinadas a diferentes puertos. Para estas situaciones, es mejor utilizar *marcas de cortafuegos*.

1.8.4.2. Marcas de cortafuegos

Las marcas de cortafuegos ofrecen una manera fácil y eficiente de agrupar puertos utilizados por un protocolo o grupo de protocolos relacionados. Por ejemplo, si LVS se implementa en un sitio de comercio electrónico, las marcas de cortafuegos pueden ser usadas para agrupar conexiones HTTP en el puerto 80 y conexiones seguras en el puerto 443. Al asignar la misma marca de cortafuego al

servidor virtual para cada protocolo, la información de estado para la transacción puede ser preservada porque el enrutador LVS envía todas las solicitudes al mismo servidor real después de que la conexión ha sido abierta.

Gracias a su eficiencia y facilidad de uso, los administradores de LVS deben utilizar marcas de cortafuegos en vez de persistencia cuando sea posible para agrupar conexiones. Sin embargo, se debe añadir persistencia a los servidores virtuales junto a las marcas de cortafuegos para asegurar que los clientes se reconecten al mismo servidor por un periodo de tiempo adecuado.

1.9. HERRAMIENTAS DE ADMINISTRACIÓN DE CLUSTER

Red Hat Cluster Suite proporciona una variedad de herramientas para configurar y administrar su cluster de Red Hat. Esta sección proporciona un resumen de las herramientas de administración disponibles con Red Hat Cluster Suite:

- Sección 1.9.1, “Conga”
- Sección 1.9.2, “Interfaz gráfica de administración de cluster”
- Sección 1.9.3, “Herramientas de administración desde la línea de comandos”

1.9.1. Conga

Conga es un conjunto integrado de componentes de software que proporciona tareas de configuración y administración centralizada para los cluster y el almacenamiento de Red Hat. **Conga** ofrece las siguientes funcionalidades:

- Interfaz de web para administrar cluster y almacenaje
- Implementación automatizada de los datos del cluster y paquetes de soporte
- Integración fácil con los cluster existentes
- No hay necesidad de reautenticación
- Integración de los registros y estado del cluster
- Control detallado sobre los permisos de usuarios

Los componentes primarios de **Conga** son **luci** y **ricci**, los cuales pueden ser instalados por separado. **luci** es un servidor que se ejecuta en un computador y se comunica con varios cluster y computadores a través de **ricci**. **ricci** es un agente que se ejecuta en cada computador (ya sea un miembro de un cluster o un computador independiente) administrado por **Conga**.

Se puede acceder a **luci** a través de un navegador de Web. Este proporciona tres funcionalidades principales a las cuales se puede acceder a través de las siguientes pestañas:

- **homebase** — Proporciona herramientas para añadir y borrar computadores, añadir y borrar usuarios y configurar privilegios de usuarios. Sólo un administrador de sistema puede acceder a esta pestaña.
- **cluster** — Proporciona herramientas para crear y configurar los cluster. Cada instancia de **luci** lista los cluster que han sido establecidos con esa instancia de **luci**. Un administrador de sistema puede administrar todos los cluster listados en esta pestaña. Otros usuarios pueden administrar solo los cluster a los cuales tiene permiso de administrar (otorgados por el administrador).

- **storage** — proporciona herramientas para la administración remota del almacenamiento. Con las herramientas en esta pestaña, usted puede administrar el almacenaje en computadores (sin importar si estos pertenecen o no al cluster).

Para administrar un cluster o almacenaje, un administrador añade (o *registra*) un cluster o un computador a un servidor **luci**. Cuando un cluster o computador es registrado con **luci**, el nombre de host del nombre de dominio completamente calificado o la dirección IP de cada computador se almacena en la base de datos **luci**.

Puede poblar la base de datos de una instancia de **luci** desde otra instancia de **luci**. Esta funcionalidad proporciona un medio de replicar a un servidor **luci** y proporciona una ruta de prueba y actualización eficiente. Cuando instala una instancia de **luci**, la base de datos está vacía. Sin embargo, usted puede importar parte o toda la base de datos **luci** de un servidor **luci** existente al implementar un nuevo servidor **luci**.

Cada instancia de **luci** tiene un usuario durante la instalación inicial — admin. Solo el usuario admin puede añadir sistemas al servidor **luci**. Asimismo, el usuario de administración puede crear cuentas de usuario adicionales y determinar cuales usuarios pueden tener acceso a los cluster y servidores registrados en la base de datos de **luci**. Es posible importar varios usuarios, cluster y computadores en una sola operación en un nuevo servidor **luci**

Cuando un computador se añade al servidor **luci** para que este sea administrado, la autenticación se realiza una sola vez. No hay necesidad de realizar más autenticaciones (a menos que el certificado utilizado sea revocado por un CA). Después de esto, puede configurar y administrar cluster y almacenamiento de forma remota a través de la interfaz de usuario **luci**. **luci** y **ricci** se comunican usando XML.

Las siguientes figuras muestran ejemplos de las tres principales pestañas de **luci**: **homebase**, **cluster** y **storage**.

Para mayor información sobre **Conga**, consulte *Configurando y administrando un cluster de Red Hat* y la ayuda en línea disponible en el servidor **luci**.

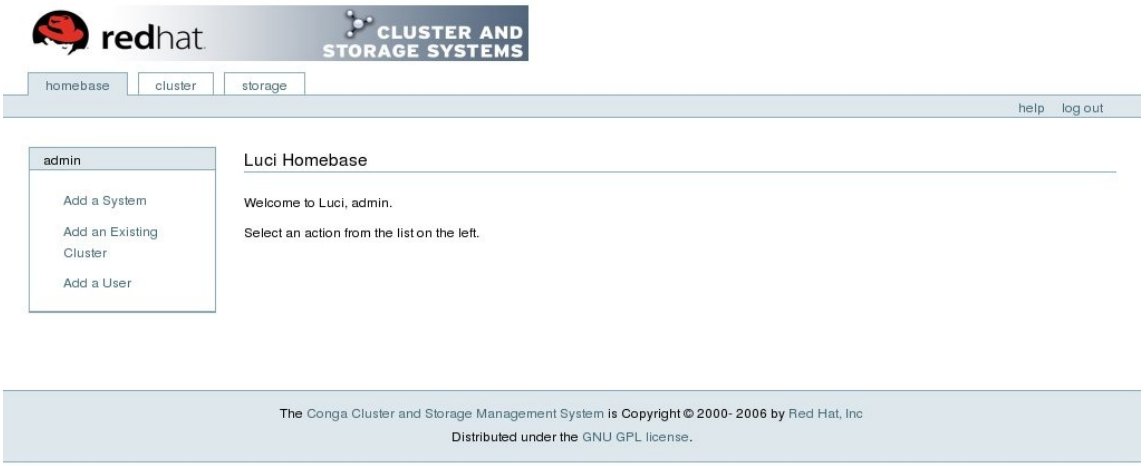


Figura 1.24. Pestaña homebase de luci

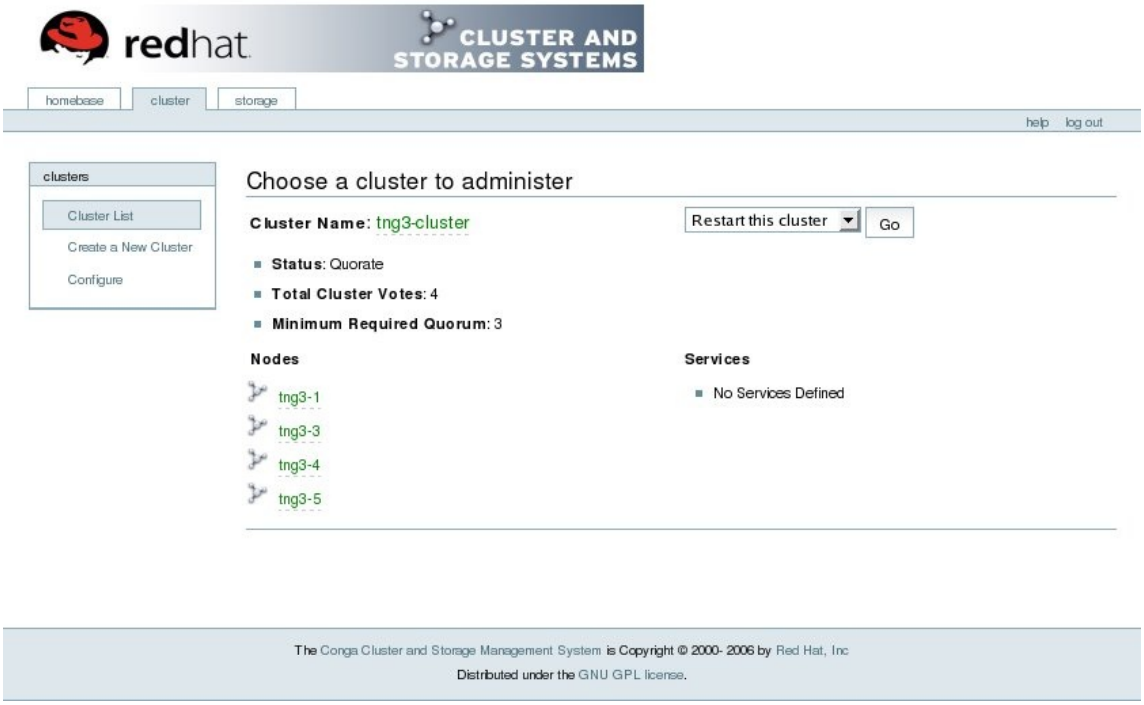


Figura 1.25. Pestaña cluster de luci

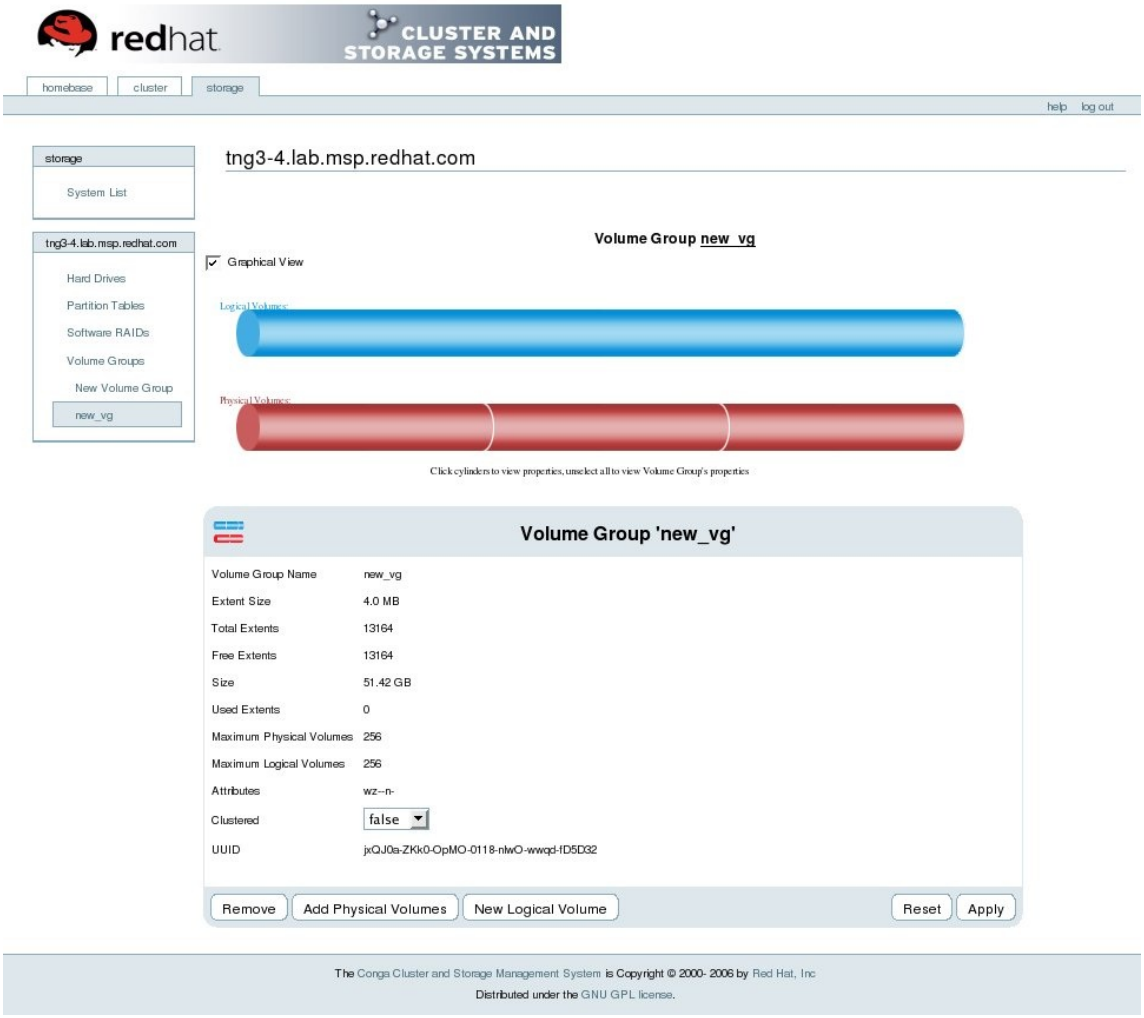


Figura 1.26. Pestaña storage de luci

1.9.2. Interfaz gráfica de administración de cluster

This section provides an overview of the **system-config-cluster** cluster administration graphical user interface (GUI) available with Red Hat Cluster Suite. The GUI is for use with the cluster infrastructure and the high-availability service management components (refer to Sección 1.3, "Cluster Infrastructure" and Sección 1.4, "Administración de servicios de alta disponibilidad"). The GUI consists of two major functions: the **Cluster Configuration Tool** and the **Cluster Status Tool**. The **Cluster Configuration Tool** provides the capability to create, edit, and propagate the cluster configuration file (**/etc/cluster/cluster.conf**). The **Cluster Status Tool** provides the capability to manage high-availability services. The following sections summarize those functions.

- Sección 1.9.2.1, "**Cluster Configuration Tool**"
- Sección 1.9.2.2, "**Cluster Status Tool**"

1.9.2.1. Cluster Configuration Tool

You can access the **Cluster Configuration Tool** (Figura 1.27, "**Cluster Configuration Tool**") through the **Cluster Configuration** tab in the Cluster Administration GUI.

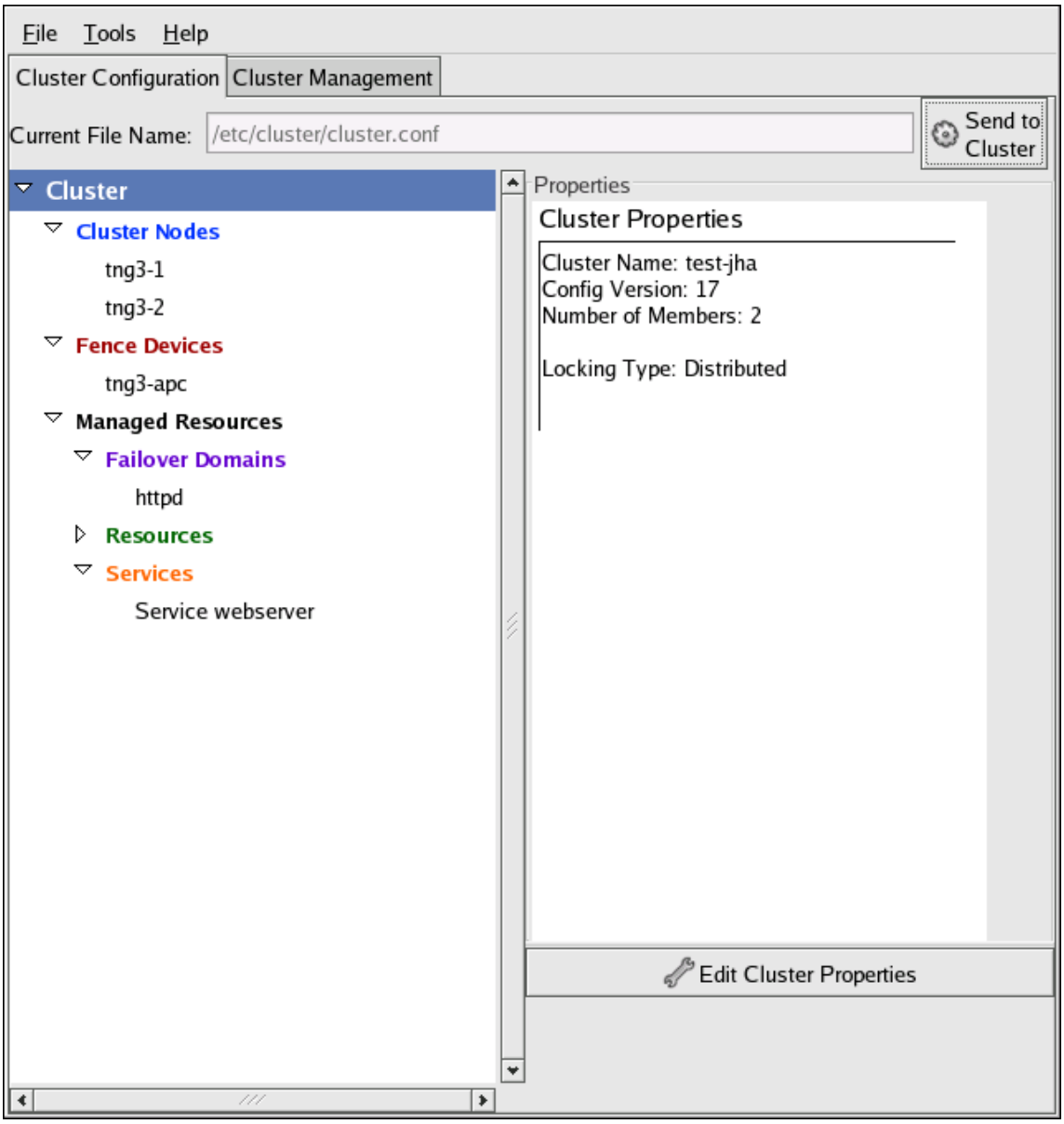


Figura 1.27. Cluster Configuration Tool

La **Cluster Configuration Tool** representa los componentes de configuración de cluster en el archivo

de configuración (**/etc/cluster/cluster.conf**) con una jerarquía gráfica que se muestra en el panel izquierdo. Un icono en forma de triángulo a la izquierda del nombre del componente indica que el componente tiene uno o más componentes subordinados asignados. Haga clic en el triángulo para expandir o cerrar la porción de árbol bajo el componente. Los componentes mostrados en la interfaz gráfica se resumen así:

- **Nodo de cluster** — Muestra los nodos de cluster. Los nodos se representan según el nombre como elementos subordinados bajo **Nodos de cluster**. Al usar los botones de configuración en la parte inferior del panel derecho (bajo **Propiedades**), usted puede añadir nodos, borrar nodos, editar nodos y configurar métodos de aislamiento para cada nodo.

- **Dispositivos de aislamiento** — Muestra los dispositivos de aislamiento. Los dispositivos de aislamiento se representan como elementos subordinados bajo **Dispositivos de aislamiento**. Con los botones de configuración en la parte inferior del panel derecho (bajo **Propiedades**), usted puede añadir dispositivos de aislamiento, borrar dispositivos de aislamiento y editar las propiedades de los dispositivos de aislamiento. Los dispositivos de aislamiento deben ser definidos antes de configurar el aislamiento (con el botón **Administrar aislamiento para este nodo**) para cada nodo.

- **Recursos administrados** — Muestra los dominios de recuperación en contra de fallos, los recursos y servicios.

  - **Dominio de recuperación** — Para configurar uno o más subgrupos de nodos de cluster utilizados para ejecutar un servicio de alta disponibilidad en caso de que un nodo falle. Los dominios de recuperación contra fallos se representan como elementos subordinados bajo **Dominios de recuperación**. Utilizando los botones de configuración en la parte inferior del panel derecho (bajo **Propiedades**), usted puede crear los dominios de recuperación contra fallos (cuando **Dominios de recuperación** está seleccionado) o editar las propiedades de un dominio de recuperación contra fallos (si el dominio está seleccionado).

  - **Recursos** — Para configurar los recursos compartidos para que sean usados por servicios de alta disponibilidad. Los recursos compartidos consisten en sistemas de archivos, direcciones IP, recursos compartidos NFS y scripts creados por el usuario que están disponibles a cualquier servicio de alta disponibilidad en el cluster. Los recursos se representan como elementos subordinados bajo **Recursos**. Con los botones ubicados en la parte inferior del panel derecho (bajo **Propiedades**), se pueden crear recursos (cuando **Recursos** está seleccionado) o editar las propiedades del recurso (cuando éste está seleccionado).

    NOTA

    La **Cluster Configuration Tool** permite también configurar los recursos privados. Un recurso privado es un recurso que es configurado para ser utilizado por un solo servicio. Se puede configurar un recurso privado con el componente **Servicio** en la interfaz gráfica.

  - **Servicios** — Para crear y configurar servicios de alta disponibilidad. Un servicio es configurado mediante la asignación de recursos (compartidos o privados), asignación de un dominio de recuperación contra fallos y la definición de una política de recuperación para el servicio. Los servicios se representan como elementos subordinados bajo **Servicios**. Con los botones ubicados en la parte inferior del panel derecho (bajo **Propiedades**), se puede crear servicios (cuando **Servicios** está seleccionado) o editar las propiedades de un servicio (cuando éste está seleccionado).

1.9.2.2. Cluster Status Tool

You can access the **Cluster Status Tool** (Figura 1.28, "**Cluster Status Tool** ") through the **Cluster Management** tab in Cluster Administration GUI.

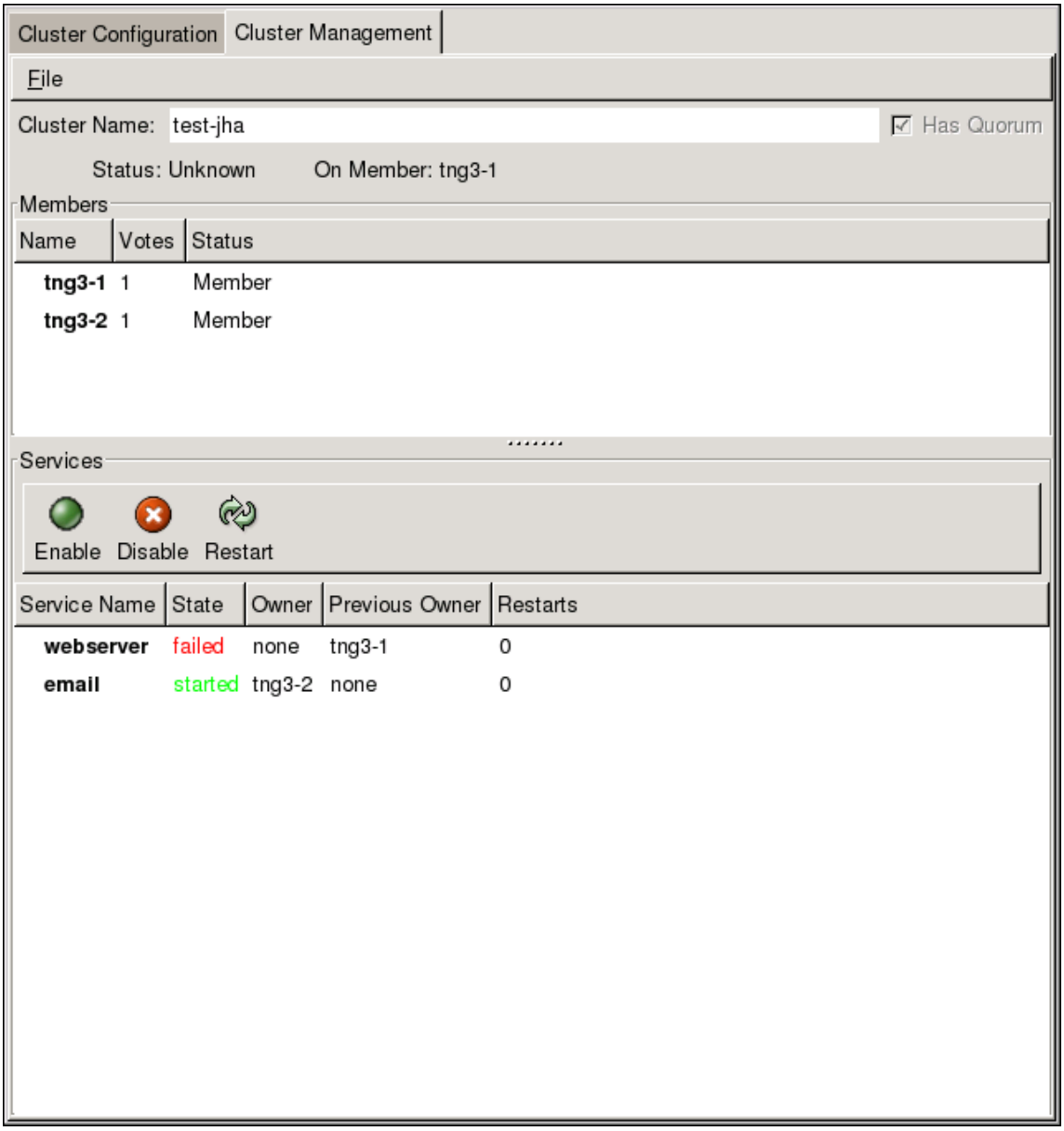


Figura 1.28. Cluster Status Tool

Los nodos y servicios mostrados en la **Cluster Status Tool** están determinados por el archivo de configuración de cluster (`/etc/cluster/cluster.conf`). Puede utilizar **Cluster Status Tool** para activar, desactivar, reiniciar o asignar los servicios de alta disponibilidad.

1.9.3. Herramientas de administración desde la línea de comandos

In addition to **Conga** and the `system-config-cluster` Cluster Administration GUI, command line tools are available for administering the cluster infrastructure and the high-availability service management components. The command line tools are used by the Cluster Administration GUI and init scripts supplied by Red Hat. Tabla 1.1, "Herramientas de la línea de comandos" summarizes the command line tools.

Tabla 1.1. Herramientas de la línea de comandos

Herramienta de la línea de comando	Usado con	Propósito
ccs_tool — Herramienta de sistema de configuración de cluster	Cluster Infrastructure	**ccs_tool** es un programa para realizar actualizaciones en línea al archivo de configuración del cluster. Con esta herramienta se puede crear y modificar componentes de la infraestructura del cluster (por ejemplo, crear un cluster o añadir y remover un nodo). Para obtener mayor información sobre esta herramienta, consulte las páginas man de ccs_tool(8).
cman_tool — Herramienta de administración de cluster	Cluster Infrastructure	**cman_tool** es un programa que maneja el administrador de cluster CMAN. Se puede utilizar para unirse o abandonar un cluster, eliminar un nodo o cambiar los votos de quórum en un nodo. Para obtener mayor información sobre esta herramienta, consulte la página de cman_tool(8).
fence_tool — Herramienta de aislamiento	Cluster Infrastructure	**fence_tool** es un programa utilizado para unirse o abandonar el dominio de aislamiento predeterminado. Específicamente, este programa inicia el daemon de aislamiento (**fenced**) para unirse al dominio y termina **fenced** para dejar el dominio. Para obtener mayor información sobre esta herramienta, consulte la página man fence_tool(8).
clustat — Utilidad de estado del cluster	Componentes de administración de servicios de alta disponibilidad	El comando **clustat** muestra el estado del cluster. Muestra la información de membresía, la vista del quórum y el estado de todos los servicios de usuario configurados. Para mayor información sobre esta herramienta consulte la página man clustat(8).
clusvcadm — Utilidad de administración de servicios de usuario del cluster	Componentes de administración de servicios de alta disponibilidad	Con el comando **clusvcadm** se pueden activar, desactivar, asignar y reiniciar los servicios de alta disponibilidad del cluster. Para mayor información sobre esta herramienta consulte la página man clusvcadm(8).

1.10. INTERFAZ GRÁFICA DE ADMINISTRACIÓN DEL SERVIDOR VIRTUAL DE LINUX

Esta sección proporciona un resumen de las herramientas de configuración LVS disponibles con Red Hat Cluster Suite — la **Piranha Configuration Tool**. La **Piranha Configuration Tool** es una interfaz de usuario gráfica basada en la web que proporciona un acercamiento estructurado para crear el archivo de configuración para LVS — **/etc/sysconfig/ha/lvs.cf**.

Para acceder a la **Piranha Configuration Tool** se necesita que el servicio **piranha-gui** esté en ejecución en el enrutador LVS activo. Puede acceder a la **Piranha Configuration Tool** de forma local o remota con un navegador de web. Se puede utilizar esta URL: **http://localhost:3636** para acceder a la interfaz de forma local. Para acceder remotamente, se puede utilizar el nombre de host o la dirección IP con **:3636**. Si está accediendo a la **Piranha Configuration Tool** de forma remota, se debe utilizar una conexión **ssh** al enrutador LVS activo como root.

Starting the **Piranha Configuration Tool** causes the **Piranha Configuration Tool** welcome page to be

displayed (refer to Figura 1.29, “The Welcome Panel”). Logging in to the welcome page provides access to the four main screens or *panels*: **CONTROL/MONITORING**, **GLOBAL SETTINGS**, **REDUNDANCY**, and **VIRTUAL SERVERS**. In addition, the **VIRTUAL SERVERS** panel contains four *subsections*. The **CONTROL/MONITORING** panel is the first panel displayed after you log in at the welcome screen.

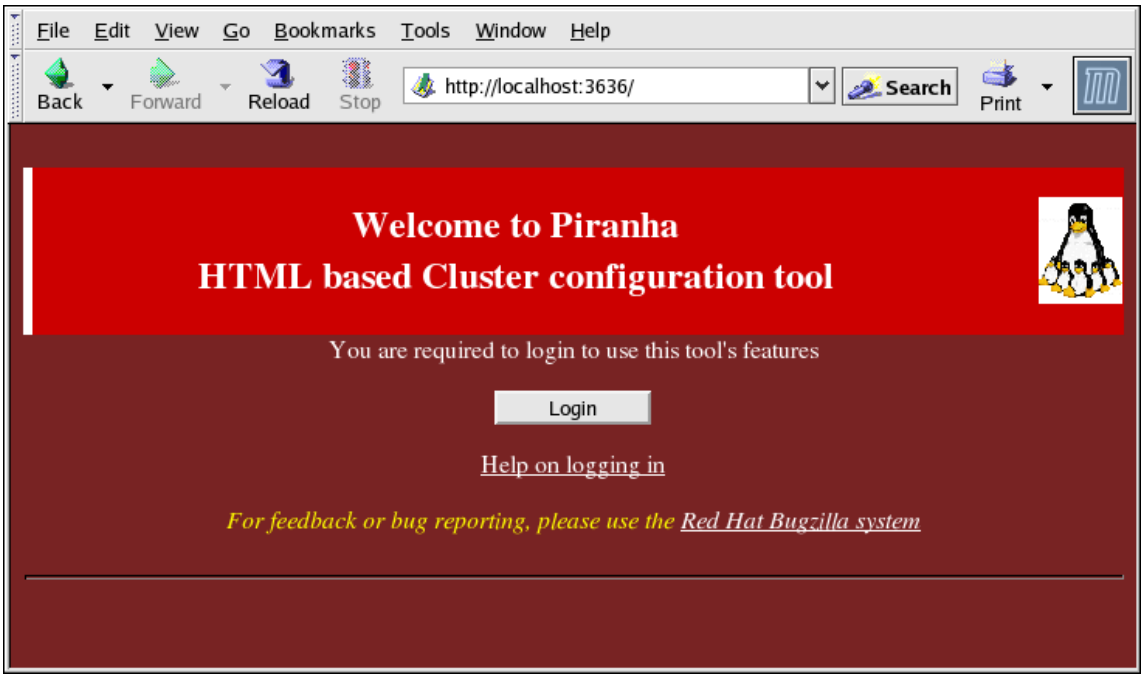


Figura 1.29. The Welcome Panel

Las siguientes secciones proporcionan una breve descripción de las páginas de configuración de la **Piranha Configuration Tool**.

1.10.1. CONTROL/MONITORING

El panel **CONTROL/MONITORING** muestra los estados en tiempo de ejecución. Muestra el estado del daemon **pulse**, la tabla de rutas de LVS y los procesos **nanny** creados por LVS.

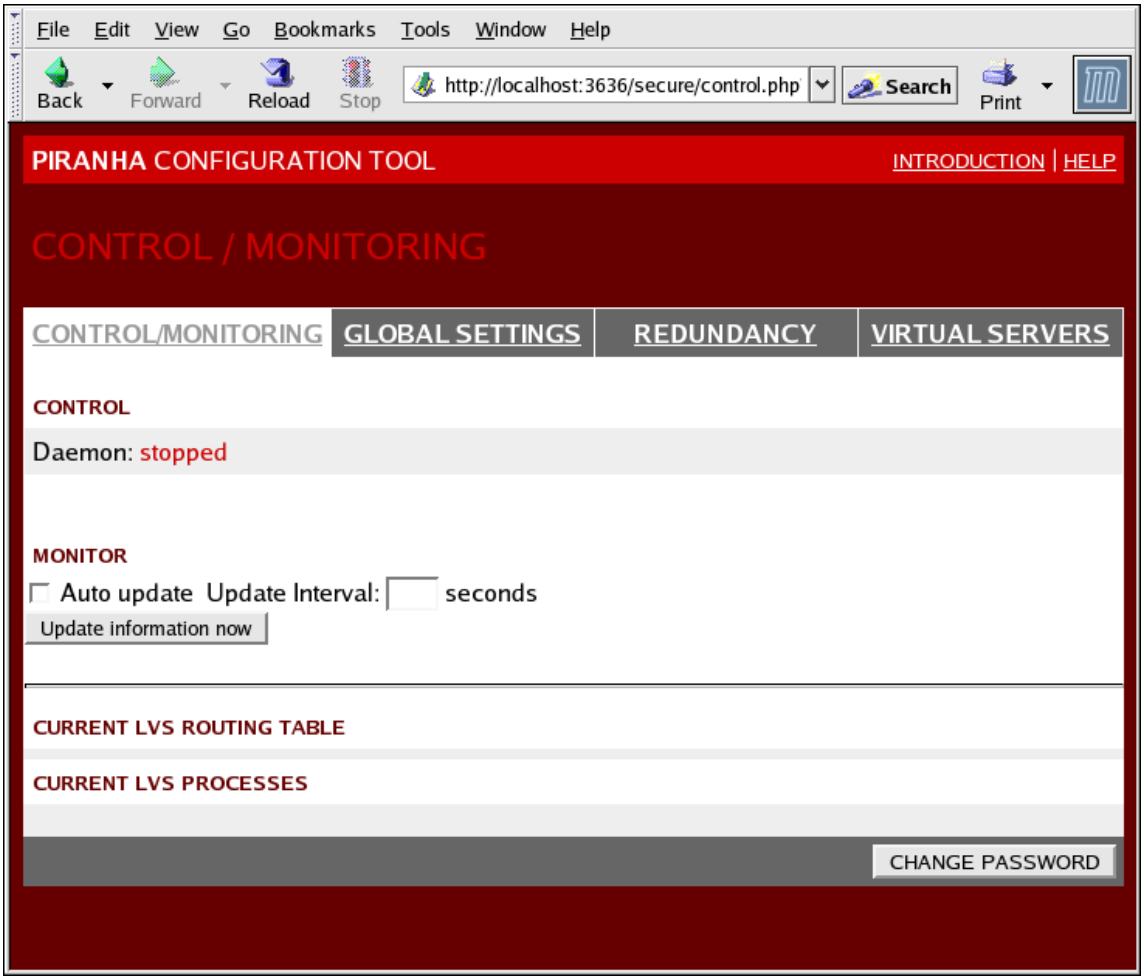


Figura 1.30. The `CONTROL/MONITORING` Panel

`Auto update`

Activa la visualización del estado para que sea actualizada de forma automática en intervalos de tiempo dados por el usuario en la casilla de texto **`Update frequency in seconds`** (el valor por defecto es 10 segundos).

No se recomienda que el intervalo de tiempo sea menor de 10 segundos. Al hacerlo, puede llegar a ser difícil reconfigurar el intervalo **`Auto update`** porque la página se actualizará con demasiada frecuencia. Si se encuentra con este problema, simplemente haga clic en otro panel y luego regrese a **`CONTROL/MONITORING`**.

`Update information now`

Proporciona la actualización manual de la información de estado.

`CHANGE PASSWORD`

Si se hace clic en este botón se tendrá acceso a una pantalla de ayuda con información sobre cómo cambiar la contraseña administrativa para la **Piranha Configuration Tool**.

1.10.2. `GLOBAL SETTINGS`

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

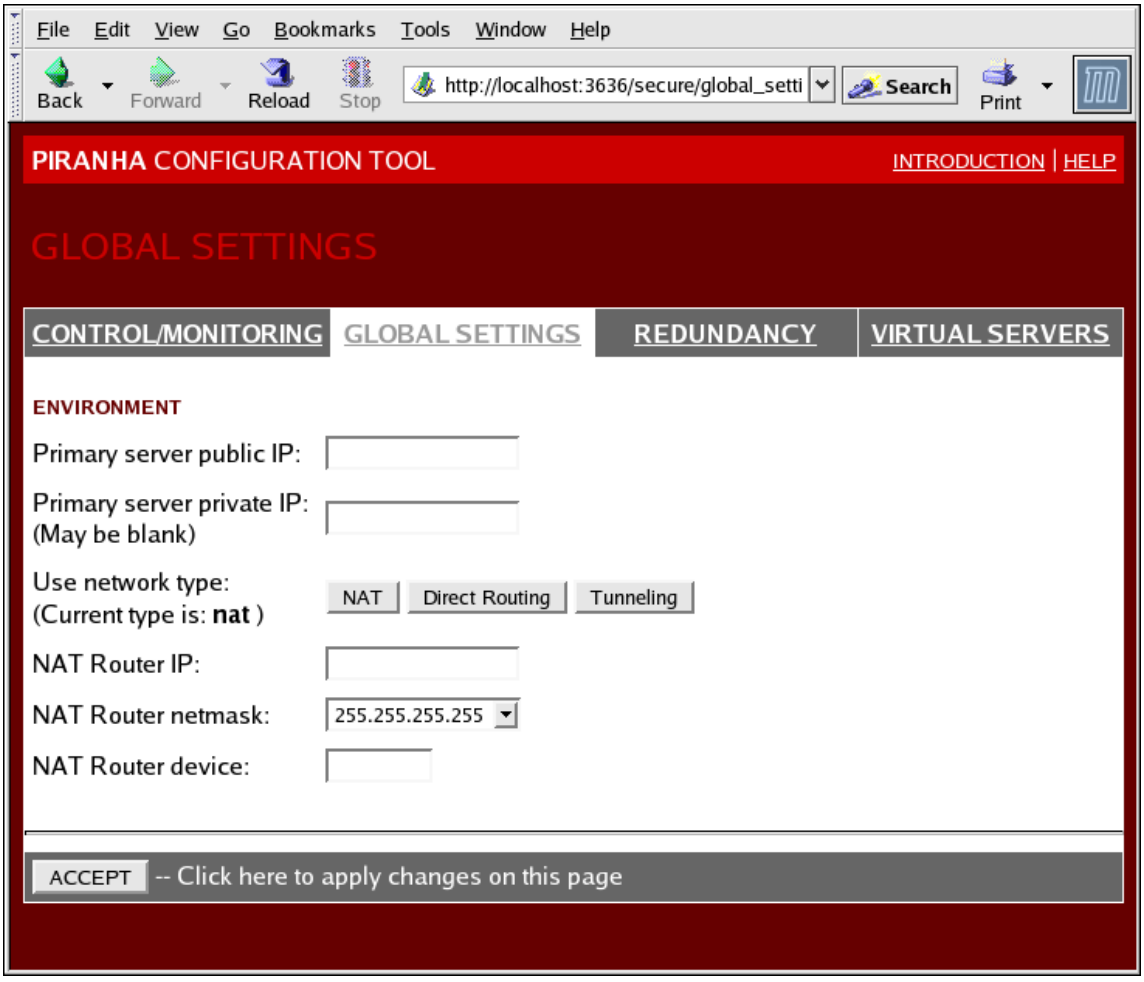


Figura 1.31. The GLOBAL SETTINGS Panel

The top half of this panel sets up the primary LVS router's public and private network interfaces.

Primary server public IP

La dirección IP real enrutable pública para el nodo LVS primario.

Primary server private IP

La dirección IP real para una interfaz de red alternativa en el nodo LVS primario. Esta dirección se utiliza únicamente como un canal alternativo de pulsos para el enrutador de respaldo.

Use network type

Selecciona el enrutado NAT

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

NAT Router IP

La IP flotante privada se define en este campo de texto. Esta IP flotante debe ser usada como puerta de enlace para los servidores reales.

NAT Router netmask

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

NAT Router device

Define el nombre del dispositivo de la interfaz de red para la dirección IP flotante, tal como **eth1:1**.

1.10.3. REDUNDANCY

El panel **REDUNDANCY** permite la configuración del enrutador LVS de respaldo y de varias opciones de sondeo de los mensajes de pulso.

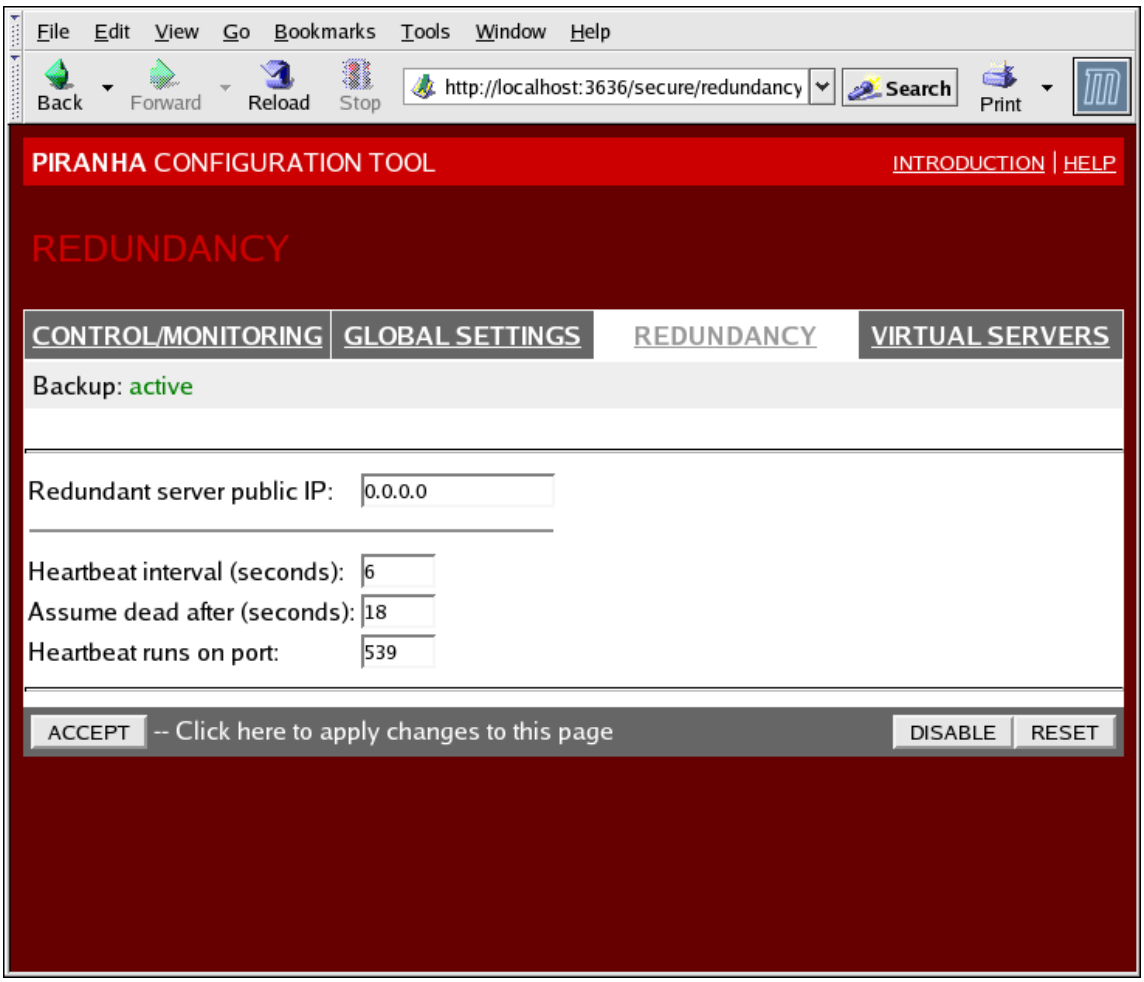


Figura 1.32. The REDUNDANCY Panel

Redundant server public IP

La dirección IP real pública para el enrutador LVS de respaldo.

Redundant server private IP

The backup router's private real IP address.

El resto del panel se utiliza para configurar el canal de pulso. El nodo de respaldo utiliza este canal para sondear la salud del nodo primario.

Heartbeat Interval (seconds)

Establece el intervalo de segundos entre pulsos — El nodo de respaldo utiliza este intervalo para revisar el estado del nodo LVS primario.

Assume dead after (seconds)

Si el nodo LVS primario no responde después de este intervalo de tiempo, el enrutador LVS de respaldo inicia el procedimiento de recuperación contra fallos.

Heartbeat runs on port

Establece el puerto utilizado para la comunicación de pulsos con el nodo LVS primario. El valor predeterminado es 539.

1.10.4. VIRTUAL SERVERS

El panel **VIRTUAL SERVERS** muestra la información para cada servidor virtual definido actualmente. Cada entrada en la tabla muestra el estado del servidor virtual, el nombre del servidor, la IP virtual asignada al servidor, la máscara de red de la IP virtual, el número de puerto en el cual el servicio se comunica, el protocolo usado y la interfaz de dispositivo virtual.

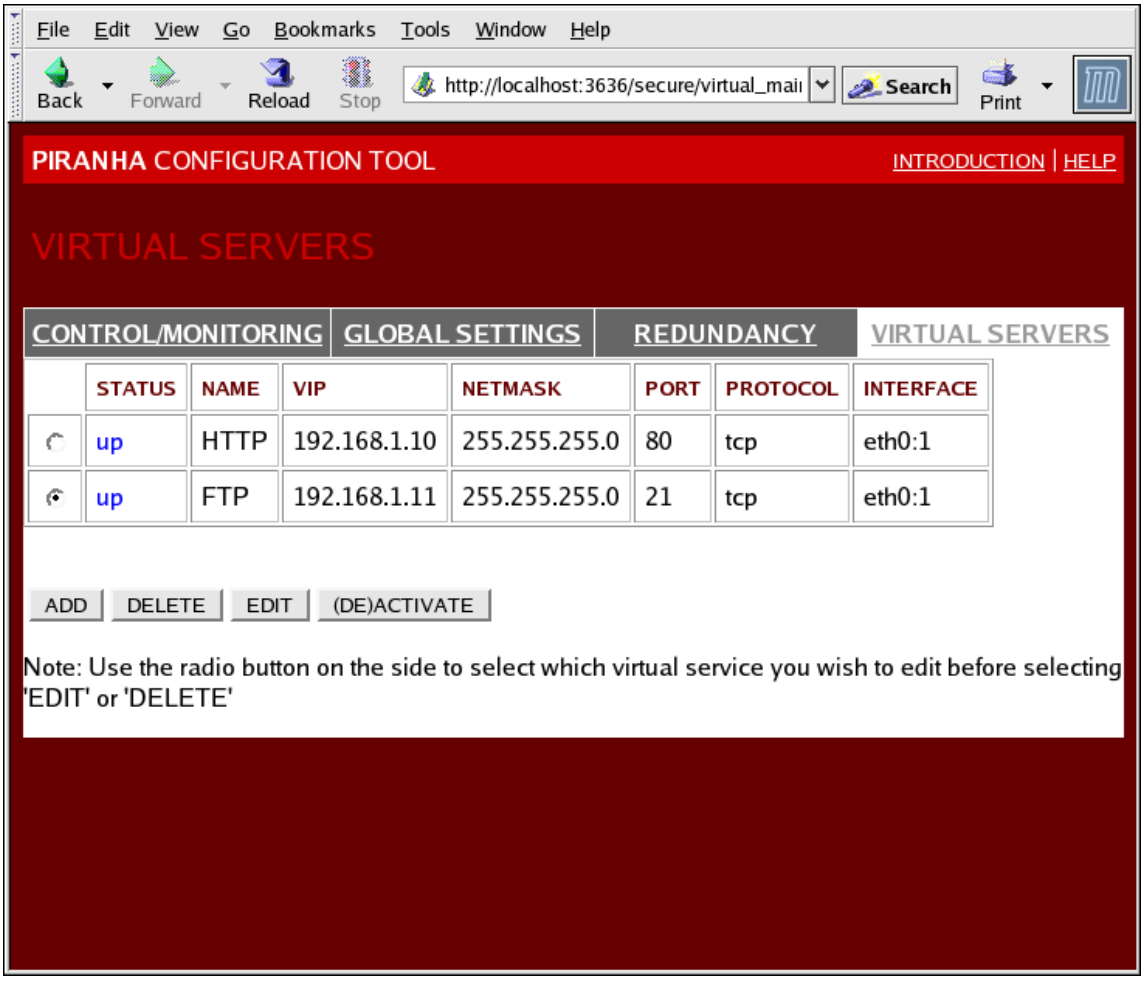


Figura 1.33. The VIRTUAL SERVERS Panel

Cada servidor mostrado en el panel **VIRTUAL SERVERS** puede ser configurado en las pantallas o *subsecciones* siguientes.

Para añadir un servicio, haga clic en el botón **ADD**. Para remover un servicio, seleccione éste haciendo clic en el botón de radio al lado del servidor virtual y luego haga clic en **DELETE**.

Para activar o desactivar un servidor virtual en la tabla, haga clic en el botón de radio apropiado y luego en el botón **(DE)ACTIVATE**.

Después de añadir un servidor virtual, éste se puede configurar si se hace clic en el botón de radio a la izquierda y luego en **EDIT** para ir a la subsección **VIRTUAL SERVER**.

1.10.4.1. La subsección VIRTUAL SERVER

The **VIRTUAL SERVER** subsection panel shown in Figura 1.34, "The **VIRTUAL SERVERS** Subsection" 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.

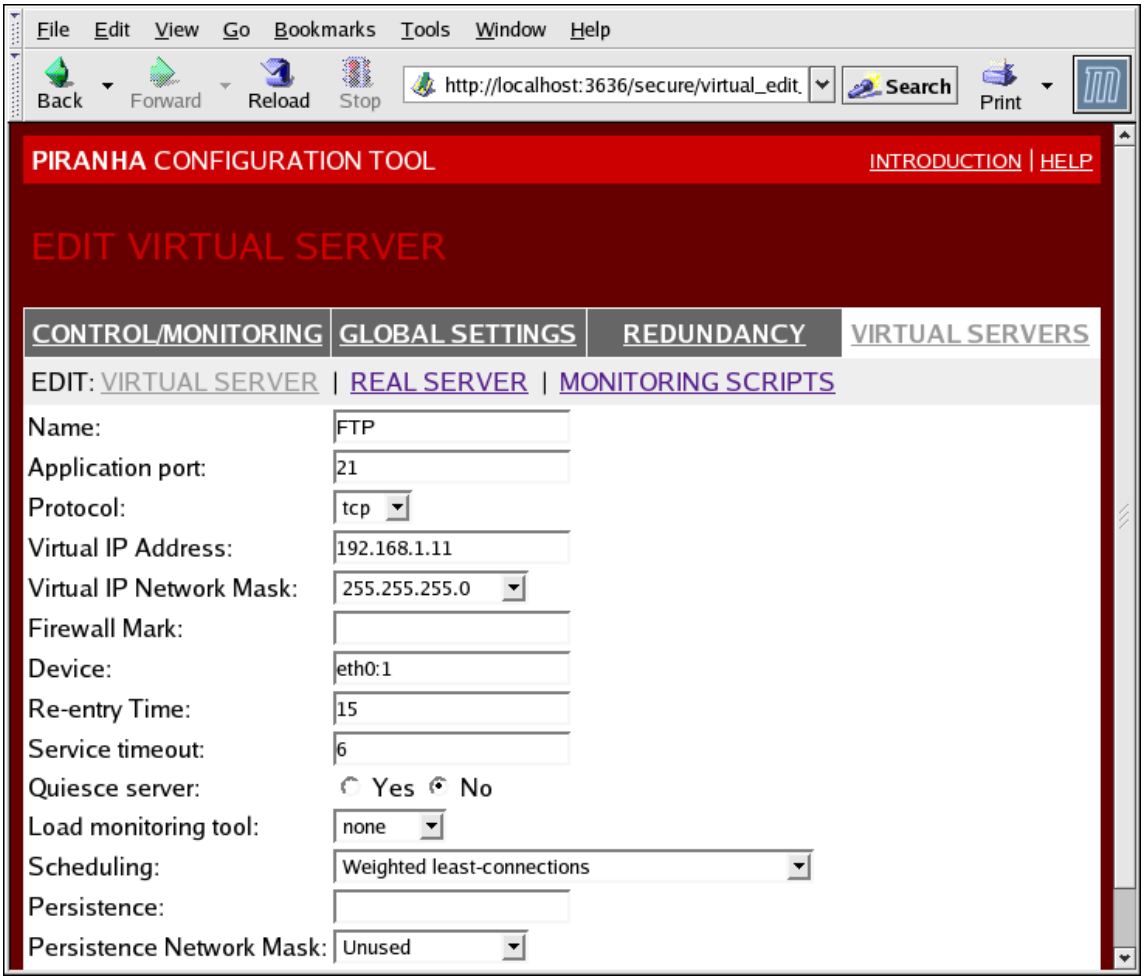


Figura 1.34. The VIRTUAL SERVERS Subsection

Name

Un nombre descriptivo para identificar el servidor virtual. Este nombre *no* es el nombre de host de la máquina, debe ser descriptivo y fácilmente identificable. Puede hacer referencia al protocolo usado por el servidor virtual (como por ejemplo HTTP).

Application port

El número de puerto a través del cual la aplicación escuchará.

Protocol

Permite elegir entre UDP y TCP.

Virtual IP Address

The virtual server's floating IP address.

Virtual IP Network Mask

La máscara de red del servidor virtual en un menú desplegable.

`Firewall Mark`

Se puede introducir un valor entero de marca de cortafuego para el agrupamiento de protocolos o para crear un servidor virtual de varios puertos por separado pero con protocolos relacionados.

`Device`

El nombre del dispositivo de red en el cual desea vincular la dirección flotante definida en el campo **`Virtual IP Address`**.

Se debe crear un alias de la dirección IP flotante a la interfaz de ethernet conectada a la red pública.

`Re-entry Time`

Un valor entero que define el número de segundos antes de que el enrutador LVS activo intente utilizar un servidor real después de que el servidor real falle.

`Service Timeout`

Un valor entero que define el número de segundos antes de que el servidor real sea considerado como no disponible.

`Quiesce server`

Si se selecciona el botón de radio **`Quiesce server`**, cada vez que un nuevo servidor entra en línea, la tabla de conexiones mínima se establece a cero para que el enrutador LVS activo enrute las solicitudes como si todos los servidores reales hubiesen sido recientemente añadidos. Esta opción previene que el nuevo servidor sea invadido por un alto número de conexiones tras entrar en el cluster.

`Load monitoring tool`

El enrutador LVS puede sondear la carga de los servidores reales utilizando **`rup`** o **`ruptime`**. Si selecciona **`rup`** desde el menú desplegable, cada servidor real debe ejecutar el servicio **`rstatd`**. Si selecciona **`ruptime`**, cada servidor real debe ejecutar el servicio **`rwhod`**.

`Scheduling`

Es el algoritmo de programación preferido. Por defecto es **`Weighted least-connection`**.

`Persistence`

Utilizado si se necesitan conexiones persistentes al servidor virtual durante las transacciones del cliente. En este campo de texto se deben especificar el número de segundos de inactividad antes de que la conexión expire.

Persistence Network Mask

Para limitar la persistencia a una subred particular, seleccione la máscara apropiada de red desde el menú desplegable.

1.10.4.2. Subsección `REAL SERVER`

Al hacer clic en el enlace de la subsección **REAL SERVER** en la parte superior del panel, se llegará a la subsección **EDIT REAL SERVER**. Muestra el estado de los hosts del servidor físico para un servicio virtual particular.

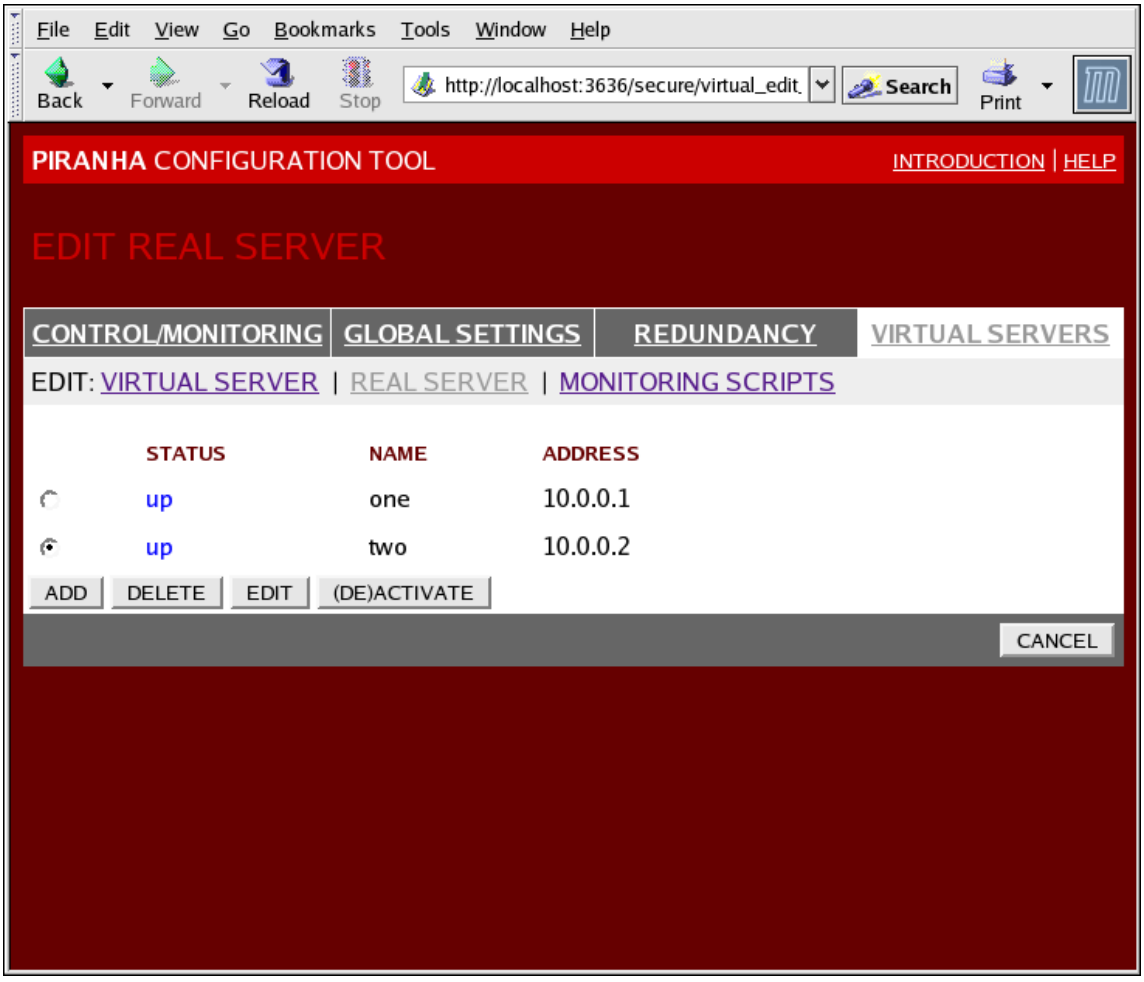


Figura 1.35. The REAL SERVER Subsection

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 Figura 1.36, "The **REAL SERVER** Configuration Panel".

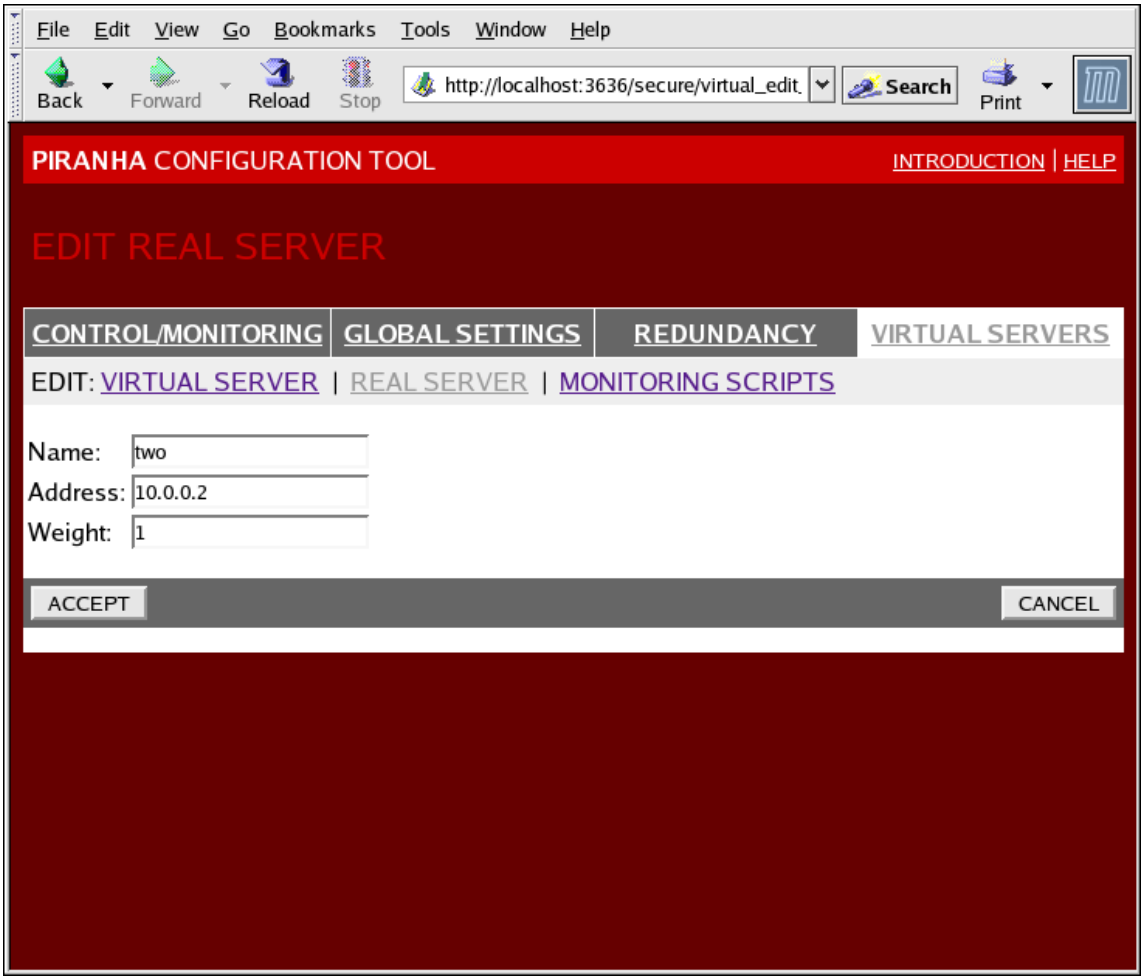


Figura 1.36. The REAL SERVER Configuration Panel

Este panel está constituido por tres campos:

Name

Un nombre descriptivo para el servidor real.

NOTA

Este nombre *no* es el nombre de host de la máquina. Utilice un nombre descriptivo y fácilmente identificable.

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.

1.10.4.3. EDIT MONITORING SCRIPTS Subsection

Haga clic en el enlace **MONITORING SCRIPTS** en la parte superior de la página. La subsección **EDIT MONITORING SCRIPTS** permite que los administradores especifiquen una secuencia de envío y

expectativa para verificar que el servicio para el servidor virtual esté funcionando en cada servidor real. También es posible especificar scripts personalizados para revisar los servicios que requieren cambios de datos de forma dinámica.

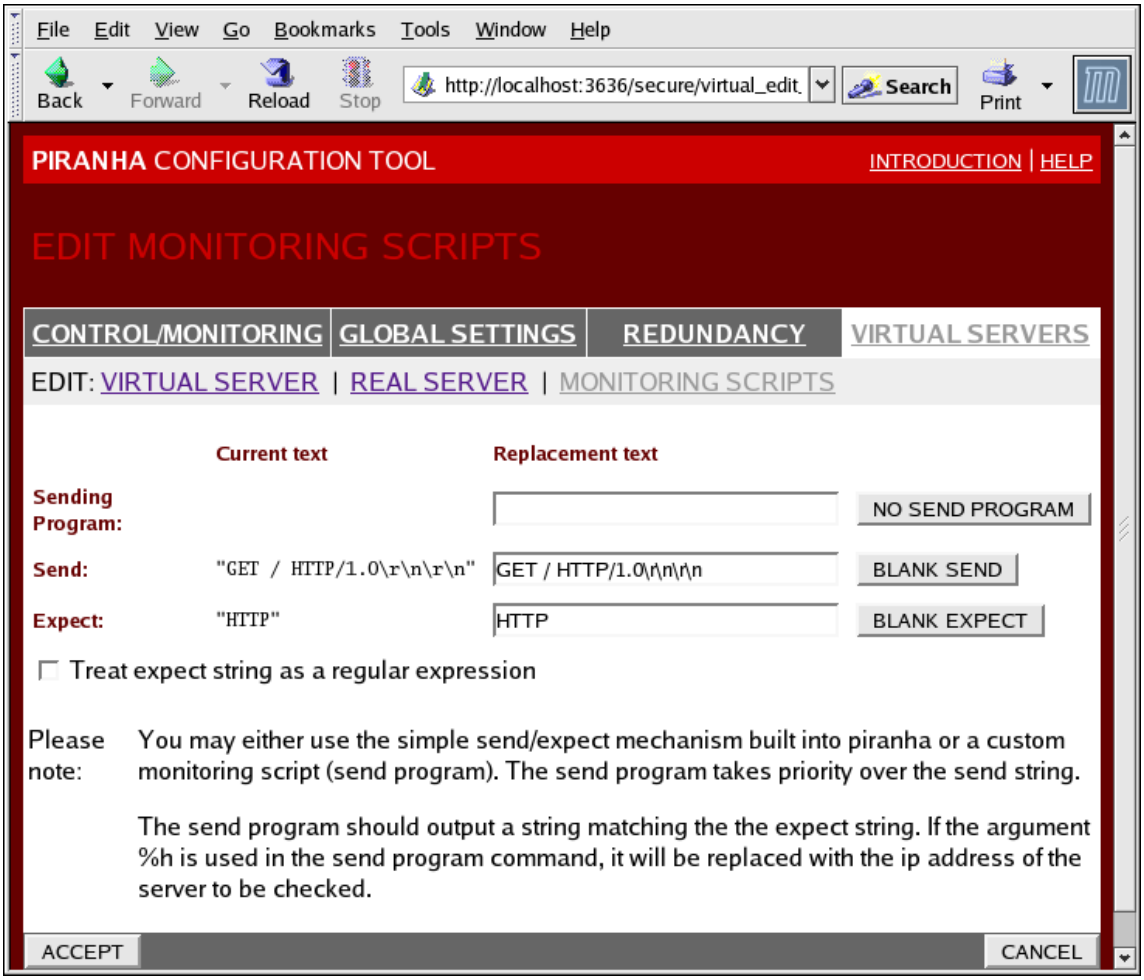
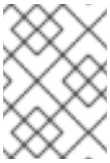


Figura 1.37. The `EDIT MONITORING SCRIPTS` Subsection

`Sending Program`

Se puede utilizar este campo para especificar un script para una verificación de servicios más avanzada. Esta función es especialmente útil para servicios que requieren cambios de datos de forma dinámica, como HTTPS o SSL.

Para usar esta función, se debe escribir un script que retorne una respuesta textual. El script debe ser ejecutable y su ruta debe establecerse en el campo **`Sending Program`**.



NOTA

Si se introduce un programa externo en el campo **`Sending Program`**, el campo **Send** será ignorado.

Send

Una cadena para el daemon **nanny** que será enviada a cada servidor real. Por defecto la entrada se completa para HTTP. Se puede alterar este valor dependiendo de sus necesidades. Si se deja este campo en blanco, el daemon **nanny** intentará abrir el puerto y, si lo logra, asumirá que el servicio está en ejecución.

Solo una secuencia de envío es permitida en este campo y solo puede contener caracteres ASCII y los siguientes caracteres de escape:

- \n para nueva línea.
- \r para retorno de línea.
- \t para tablatura.
- \ para escapar el siguiente caracter.

Expect

La respuesta textual que el servidor debe dar si está funcionando apropiadamente. Si escribió su propio programa de envío, introduzca la respuesta esperada.

---

[1] Un servidor virtual es un servicio configurado para escuchar por una IP virtual específica.

CAPÍTULO 2. RESUMEN DE COMPONENTES DE RED HAT CLUSTER SUITE

Este capítulo proporciona un resumen de los componentes de Red Hat Cluster Suite. Este capítulo tiene las siguientes secciones:

- Sección 2.1, “Componentes de cluster”
- Sección 2.2, “Páginas de manual (man)”
- Sección 2.3, “Compatibilidad de hardware”

2.1. COMPONENTES DE CLUSTER

Tabla 2.1, “Componentes del subsistema de software Red Hat Cluster Suite” summarizes Red Hat Cluster Suite components.

Tabla 2.1. Componentes del subsistema de software Red Hat Cluster Suite

Función	Componentes	Descripción
Conga	`luci`	Sistema de administración remota - estación de administración.
	`ricci`	Sistema de administración remota - estación administrada.
Cluster Configuration Tool	`system-config-cluster`	Comando utilizado para administrar la configuración del cluster en un entorno gráfico.
Administrador de volúmenes lógicos de cluster (CLVM)	`clvmd`	El daemon que distribuye las actualizaciones de metadatos LVM en un cluster. Se debe ejecutar en todos los nodos del cluster. Si un nodo no está ejecutando este daemon se reportará un error.
	`lvm`	Herramientas LVM2. Proporciona las herramientas para la línea de comandos para LVM2.
	`system-config-lvm`	Proporciona una interfaz gráfica para LVM2.
	`lvm.conf`	El archivo de configuración de LVM. La ruta completa es `/etc/lvm/lvm.conf`.

Función	Componentes	Descripción
Sistema de configuración del Cluster (CCS)	**ccs_tool**	**ccs_tool** es parte del sistema de configuración del cluster (CCS). Se utiliza para hacer actualizaciones en línea de los archivos de configuración de CCS. Además puede ser utilizado para actualizar los archivos de configuración de cluster desde los archivos creados por GFS 6.0 (o anterior) al formato en XML utilizado en este lanzamiento de Red Hat Cluster Suite.
	ccs_test	Comando de diagnóstico y prueba utilizado para obtener información desde los archivos de configuración a través de **ccsd**.
	ccsd	El daemon CCS que es ejecutado en todos los nodos de cluster y proporciona los datos del archivo de configuración al software de cluster.
	cluster.conf	Este es el archivo de configuración del cluster. La ruta completa es **/etc/cluster/cluster.conf**.
Administrador de cluster (CMAN)	**cman.ko**	El módulo de kernel para CMAN.
	cman_tool	Esta es la interfaz administrativa para CMAN. Inicia y detiene CMAN y puede cambiar algunos parámetros internos (por ejemplo los votos).
	dlm_controld	Daemon iniciado por el script de inicio de **cman** para administrar **dlm** en el kernel. No utilizado por el usuario.
	gfs_controld	Daemon iniciado por el script de inicio de **cman** para administrar **gfs** en el kernel. No utilizado por el usuario.
	group_tool	Utilizado para obtener una lista de los grupos relacionados con el proceso de aislamiento, DLM, GFS y obtener información de depuración; incluye funcionalidades proporcionadas por **cman_tool services** en RHEL 4.

Función	Componentes	Descripción
	groupd	Daemon iniciado por el script de inicio de **cman** para servir de interfaz entre **openais/cman** y **dlm_controld/gfs_controld/fenced**. No utilizado por el usuario.
	libcman.so.*<version number>*	Biblioteca para los programas que necesitan interactuar con **cman.ko**.
Administrador de grupos de recursos (rgmanager)	**clusvcadm**	Comando utilizado para activar, desactivar, asignar y reiniciar manualmente los servicios de usuario en un cluster
	clustat	Comando utilizado para mostrar el estado del cluster, incluyendo la membresía de los nodos y los servicios en ejecución.
	clurgmgrd	Daemon utilizado para manejar solicitudes a los servicios del usuario incluyendo el inicio, desactivación, asignación y reinicio de éstos.
	clurmtabd	Daemon utilizado para manejar tablas de montaje NFS en clusters.
Aislamiento	**fence_apc**	Agente de aislamiento para interruptores de energía APC.
	fence_bladecenter	Agente blende para IBM Bladecenters con interfaz Telnet.
	fence_bullpap	Agente de aislamiento para la interfaz Bull Novascale Platform Administration Processor (PAP).
	fence_drac	Agente de aislamiento para las tarjetas de acceso remoto de Dell
	fence_ipmilan	Agente de aislamiento para máquinas controladas por IPMI (Intelligent Platform Management Interface) sobre una LAN.
	fence_wti	Agente de aislamiento para el interruptor de energía WTI.
	fence_brocade	Agente de aislamiento para el interruptor de canal de fibra Brocade.

Función	Componentes	Descripción
	fence_mcdata	Agente de aislamiento para el interruptor de canal de fibra McData.
	fence_vixel	Agente de aislamiento para el interruptor de canal de fibra Vixel.
	fence_sanbox2	Agente de aislamiento para el interruptor de canal de fibra SANBox2.
	fence_ilo	Agente de aislamiento para las interfaces HP ILO (anteriormente fence_rib).
	fence_rsa	Agente de aislamiento de E/S para IBM RSA II.
	fence_gnbd	Agente de aislamiento utilizado con almacenamiento GNDB.
	fence_scsi	Agente de aislamiento de E/S para las reservaciones SCSI persistentes.
	fence_egenera	Agente de aislamiento utilizado con sistemas Egenera BladeFrame.
	fence_manual	Agente de aislamiento para interacción manual. *NOTA* Este componente no está soportado en entornos de producción.
	fence_ack_manual	Interfaz de usuario para el agente **fence_manual**.
	fence_node	Un programa que ejecuta procesos de aislamiento de E/S en un solo nodo.
	fence_xvm	Agente de aislamiento de E/S para las máquinas virtuales Xen.
	fence_xvmd	Agente anfitrión del proceso de aislamiento de E/S para máquinas virtuales Xen.
	fence_tool	Un programa para unirse o separarse de un dominio de aislamiento.
	fenced	El daemon de aislamiento de E/S.
DLM	**libdlm.so.<*version number*>**	Biblioteca para el soporte de DLM (siglas en inglés de Distributed Lock Manager)

Función	Componentes	Descripción
GFS	**gfs.ko**	Módulo del kernel que implementa el sistema de archivos GFS y se carga en los nodos de cluster GFS.
	gfs_fsck	Comando que repara un sistema de archivo GFS no montado.
	gfs_grow	Comando que incrementa un sistema de archivo GFS montado.
	gfs_jadd	Comando que añade el registro por diario (journal) en un sistema de archivo GFS.
	gfs_mkfs	Comando que crea un sistema de archivos GFS en un dispositivo de almacenaje.
	gfs_quota	Comando que administra quotas en un sistema de archivos GFS montado.
	gfs_tool	Comando que configura o sintoniza un sistema de archivos GFS. Este comando puede también obtener información variada sobre el sistema de archivos.
	mount.gfs	Ayudante de montaje que es llamado por **mount(8)**; no utilizado por el usuario.
GNBD	**gnbd.ko**	Módulo de kernel que implementa el controlador de dispositivos GNBD en clientes.
	gnbd_export	Comando para crear, exportar y administrar GNBDs en un servidor GNBD.
	gnbd_import	Comando para importar y administrar GNBDs en un cliente GNBD.
	gnbd_serv	Un daemon de servidor que le permite a un nodo exportar el almacenamiento local a través de la red.

Función	Componentes	Descripción
LVS	**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 LVS 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 LVS router fails to respond after a user-configurable interval, it initiates failover. During failover, **pulse** on the backup LVS router instructs the **pulse** daemon on the active LVS router to shut down all LVS services, starts the **send_arp** program to reassign the floating IP addresses to the backup LVS router's MAC address, and starts the **lvs** daemon.
	lvsd	El daemon **lvs** es ejecutado en el enrutador LVS activo una vez es llamado por **pulse**. Lee el archivo de configuración **/etc/sysconfig/ha/lvs.cf**, llama a la utilidad **ipvsadm** para construir y mantener la tabla de rutas IPVS y asignar un proceso **nanny** para cada servicio LVS configurado. Si **nanny** reporta que un servidor real ha sido apagado, **lvs** ordena a la utilidad **ipvsadm** remover el servidor real de la tabla de rutas IPVS.
	ipvsadm	Este servicio actualiza la tabla de rutas IPVS en el kernel. El daemon **lvs** configura un administrador LVS llamando **ipvsadm** para añadir o borrar entradas en la tabla de rutas IPVS.
	nanny	El daemon de sondeo **nanny** es ejecutado en el enrutador LVS activo. A través de este daemon, el enrutador LVS activo determina el estado de cada servidor real y,opcionalmente, sondea sus cargas de trabajo. Se ejecuta un proceso separado para cada servicio definido en cada servidor real.

Función	Componentes	Descripción
	lvs.cf	Este es el archivo de configuración LVS. La ruta completa del archivo es **/etc/sysconfig/ha/lvs.cf**. Directa o indirectamente, todos los daemons obtienen la información de configuración desde este archivo.
	Piranha Configuration Tool	Esta es la herramienta de web para monitorizar, configurar y administrar LVS. Ésta es la herramienta predeterminada para mantener el archivo de configuración LVS **/etc/sysconfig/ha/lvs.cf**
	send_arp	Este programa envía señales ARP cuando la dirección IP de punto flotante cambia de un nodo a otro durante el proceso de recuperación contra fallos.
Disco quórum	**qdisk**	Un daemon quórum basado en disco para Cluster de Linux / CMAN.
	mkqdisk	Utilidad de disco quórum de cluster
	qdiskd	Daemon de disco quórum de cluster

2.2. PÁGINAS DE MANUAL (MAN)

Esta sección lista las páginas de manual (man) que son relevantes a Red Hat Cluster Suite como recurso adicional.

- Infraestructura de cluster
  - ccs_tool (8) - La herramienta para realizar actualizaciones en línea de los archivos de configuración CSS
  - ccs_test (8) - La herramienta de diagnóstico para ejecutar el sistema de configuración de cluster
  - ccsd (8) - El daemon usado para acceder a los archivos de configuración de cluster CCS
  - ccs (7) - Sistema de configuración de cluster
  - cman_tool (8) - Herramienta de administración de cluster
  - cluster.conf [cluster] (5) - El archivo de configuración para los productos de cluster
  - qdisk (5) - un daemon quórum basado en disco para Cluster de Linux / CMAN
  - mkqdisk (8) - Utilidad de disco quórum de cluster
  - qdiskd (8) - daemon de disco quórum de cluster

- fence_ack_manual (8) - programa ejecutado como operador como parte de la operación de aislamiento de E/S manual
- fence_apc (8) - agente de aislamiento de E/S para APC MasterSwitch
- fence_bladecenter (8) - agente de aislamiento de E/S para IBM Bladecenter
- fence_brocade (8) - Agente de proceso de aislamiento de E/S para interruptores Brocade FC
- fence_bullpap (8) - agente de aislamiento de E/S para la arquitectura Bull FAME controlada por una consola de administración PAP
- fence_drac (8) - agente de aislamiento para las tarjetas de acceso remoto de Dell
- fence_egenera (8) - agente de aislamiento de E/S para Egenera BladeFrame
- fence_gnbd (8) - agente de aislamiento de E/S para clusters GFS basados en GNBD
- fence_ilo (8) - agente de aislamiento de E/S para las tarjetas HP Integrated Lights Out
- fence_ipmilan (8) - agente de aislamiento de E/S para máquinas controladas por IPMI sobre LAN
- fence_manual (8) - programa ejecutado por fenced como parte de las operaciones de aislamiento de E/S manual
- fence_mcdata (8) - agente de aislamiento de E/S para los interruptores de canal de fibra McData
- fence_node (8) - un programa que ejecuta operaciones de aislamiento de E/S en un nodo único
- fence_rib (8) - agente de aislamiento de E/S para tarjetas Compaq Remote Insight Lights Out
- fence_rsa (8) - agente de aislamiento de E/S para IBM RSA II
- fence_sanbox2 (8) - agente de aislamiento de E/S para interruptores de canal de fibra QLogic SANBox2
- fence_scsi (8) - agente de aislamiento de E/S para las reservaciones SCSI persistentes
- fence_tool (8) - un programa para unirse o separarse de un dominio de aislamiento
- fence_vixel (8) - Agente de aislamiento de E/S para interruptores de canal de fibra Vixel
- fence_wti (8) - agente de aislamiento de E/S para interruptores de energía de red WTI
- fence_xvm (8) - agente de aislamiento de E/S para las máquinas virtuales Xen
- fence_xvmd (8) - agente anfitrión de aislamiento de E/S para máquinas virtuales Xen
- fenced (8) - el daemon de aislamiento de E/S

- Administración de servicios de alta disponibilidad
  - clusvcadm (8) - utilidad de administración de servicios de usuario del cluster

- clustat (8) - utilidad de estado del cluster
- Clurgmgrd [clurgmgrd] (8) - daemon administrador de grupos de recursos (servicio de cluster)
- clurmtabd (8) - daemon de tabla de montaje remota NFS de cluster

- GFS
  - gfs_fsck (8) - corrector del sistema de archivos GFS fuera de línea
  - gfs_grow (8) - expande un sistema de archivos GFS
  - gfs_jadd (8) - añade el registro por diario (journal) a un sistema de archivos GFS
  - gfs_mount (8) - opciones de montaje GFS
  - gfs_quota (8) - manipula las cuotas en los discos GFS
  - gfs_tool (8) - interfaz para llamadas ioctl de gfs
- Administrador de volúmenes lógicos de cluster
  - clvmd (8) - daemon LVM de cluster
  - lvm (8) - herramientas LVM2
  - lvm.conf [lvm] (5) - archivo de configuración para LVM2
  - lvmchange (8) - cambia los atributos del administrador de volúmenes lógicos
  - pvcreate (8) - inicializa un disco o partición para ser usado por LVM
  - lvs (8) - reporta información sobre los volúmenes lógicos
- Dispositivo de bloque de red global (GNBD)
  - gnbd_export (8) -la interfaz para exportar GNBDs
  - gnbd_import (8) - manipula los dispositivos de bloque GNBD en un cliente
  - gnbd_serv (8) - daemon de servidor gnbd
- LVS
  - pulse (8) - daemon de pulsos para sondear el estado de los nodos del cluster
  - lvs.cf [lvs] (5) - archivo de configuración para lvs
  - lvscan (8) - explora (todos los discos) en busca de volúmenes lógicos
  - lvsd (8) - daemon de control de los servicios de cluster de Red Hat
  - ipvsadm (8) - administración del servidor virtual de Linux
  - ipvsadm-restore (8) - restaura la tabla IPVS desde stdin
  - ipvsadm-save (8) - guarda la tabla IPVS a stdout

- nanny (8) - herramienta para monitorizar el estado de los servicios en el cluster
- send_arp (8) - herramienta para notificar a la red sobre una nueva relación entre dirección IP y dirección MAC

2.3. COMPATIBILIDAD DE HARDWARE

Para obtener información sobre el hardware que es compatible con los componentes de Red Hat Cluster Suite (por ejemplo, soporte de dispositivos de aislamiento, dispositivo de almacenamiento y interruptores de canal de fibra), consulte la guía de configuración de hardware en http://www.redhat.com/cluster_suite/hardware/

APÉNDICE A. HISTORIA DE REVISIÓN

Revisión 3-7.400	**2013-10-31**	**Rüdiger Landmann**
Rebuild with publican 4.0.0		
Revisión 3-7	**2012-07-18**	**Anthony Towns**
Rebuild for Publican 3.0		
Revisión 1.0-0	**Tue Jan 20 2008**	**Paul Kennedy**
Consolidación de Lanzamientos		

ÍNDICE