



Red Hat Enterprise Linux 5

Cluster Suite 개요

Red Hat Enterprise Linux 5 용 Red Hat Cluster Suite

역음 3

Red Hat Enterprise Linux 5 Cluster Suite 개요

Red Hat Enterprise Linux 5 용 Red Hat Cluster Suite

역음 3

Landmann

rlandmann@redhat.com

법적 공지

Copyright © 2009 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

초록

Red Hat Cluster Suite 개요에서는 Red Hat Enterprise Linux 5 용 Red Hat Cluster Suite에 대한 개요를 다루고 있습니다.

차례

소개	2
1. 피드백	2
1장. RED HAT CLUSTER SUITE 개요	4
1.1. 클러스터 기초	4
1.2. RED HAT CLUSTER SUITE INTRODUCTION	5
1.3. CLUSTER INFRASTRUCTURE	6
1.3.1. 클러스터 관리	6
1.3.2. 잠금 관리	7
1.3.3. Fencing	8
1.3.4. 클러스터 설정 시스템	11
1.4. 고가용성 서비스 관리	13
1.5. RED HAT GFS	15
1.5.1. 우수한 성능 및 확장성	16
1.5.2. 성능, 확장성, 적절한 가격	17
1.5.3. 경제성 및 성능	18
1.6. CLVM (CLUSTER LOGICAL VOLUME MANAGER)	18
1.7. GNBD (GLOBAL NETWORK BLOCK DEVICE)	21
1.8. LVS (LINUX VIRTUAL SERVER)	22
1.8.1. Two-Tier LVS Topology	24
1.8.2. Three-Tier LVS Topology	26
1.8.3. 라우팅 방식	27
1.8.3.1. NAT 라우팅	27
1.8.3.2. 직접 라우팅	28
1.8.4. 지속성 및 방화벽 표시 기능	30
1.8.4.1. Persistence	30
1.8.4.2. 방화벽 표시 기능	30
1.9. 클러스터 관리 도구	30
1.9.1. Conga	31
1.9.2. 클러스터 관리 GUI	33
1.9.2.1. Cluster Configuration Tool	33
1.9.2.2. Cluster Status Tool	35
1.9.3. 명령행 관리 도구	36
1.10. LINUX 가상 서버 관리 GUI	37
1.10.1. CONTROL/MONITORING	38
1.10.2. GLOBAL SETTINGS	39
1.10.3. REDUNDANCY	41
1.10.4. VIRTUAL SERVERS	42
1.10.4.1. VIRTUAL SERVER 하부 섹션	43
1.10.4.2. REAL SERVER 하부 섹션	44
1.10.4.3. EDIT MONITORING SCRIPTS Subsection	46
2장. RED HAT CLUSTER SUITE 구성 요소 요약	49
2.1. 클러스터 구성 요소	49
2.2. 맨 페이지	54
2.3. 하드웨어 호환성	56
부록 A. 개정 내역	57
색인	58

소개

이 문서에서는 Red Hat Enterprise Linux 5 용 Red Hat Cluster Suite에 대한 개요를 다루고 있으며 이는 다음과 같이 구성되어 있습니다:

- 1장. *Red Hat Cluster Suite* 개요
- 2장. *Red Hat Cluster Suite* 구성 요소 요약

이 문서 내용이 대략적인 개요에 불과하더라도 Red Hat Enterprise Linux에 관한 숙련된 지식을 바탕으로 서버 컴퓨팅에 관한 개념을 이해하여 완전히 이해하시기 바랍니다.

Red Hat Enterprise Linux 사용에 관한 보다 자세한 내용은 다음 문서에서 참조하시기 바랍니다:

- *Red Hat Enterprise Linux 설치 가이드*– Red Hat Enterprise Linux 5 설치 관련 내용을 다루고 있습니다.
- *Red Hat Enterprise Linux 활용 가이드*– Red Hat Enterprise Linux 5 설정, 활용, 관리 관련 내용을 다루고 있습니다.

Red Hat Enterprise Linux 5 용 Red Hat Cluster Suite에 관한 보다 자세한 내용은 다음 문서에서 참조하시기 바랍니다:

- *Red Hat Cluster 설정 및 관리*– Red Hat Cluster 설치, 설정, 관리에 관한 내용을 다루고 있습니다.
- *Logical Volume Manager Administration*– Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- *Global File System: 설정 및 관리*– Red Hat GFS (Red Hat Global File System) 설치, 설정, 관리에 관한 내용을 다루고 있습니다.
- *Global File System 2: 설정 및 관리*– Red Hat GFS2 (Red Hat Global File System 2) 설치, 설정, 관리에 관한 내용을 다루고 있습니다.
- *Device-Mapper Multipath 사용*– Red Hat Enterprise Linux 5의 Device-Mapper Multipath 기능 사용에 관한 내용을 다루고 있습니다.
- *Global File System을 통한 GNBD 사용*– Red Hat GFS를 통한 GNBD(Global Network Block Device) 사용에 관한 개요를 다루고 있습니다.
- *Linux 가상 서버 관리*– LVS (Linux Virtual Server)를 통한 고성능 시스템 및 서비스 설정에 관한 내용을 다루고 있습니다.
- *Red Hat Cluster Suite 릴리즈 노트*– Red Hat Cluster Suite 최신 릴리즈에 관한 내용을 다루고 있습니다.

Red Hat Cluster Suite 및 기타 Red Hat 문서는 Red Hat Enterprise Linux 문서 CD에서 HTML, PDF, RPM 버전으로 보실 수 있으며 <http://www.redhat.com/docs/>에서도 보실 수 있습니다.

1. 피드백

문서 내용 개선을 위한 제안이 있거나 오자를 발견했을 경우 언제든지 알려 주시기 바랍니다.
Documentation-cluster에 대한 리포트를 버그질라(Bugzilla)에 제출해 주시기 바랍니다.
(<http://bugzilla.redhat.com/bugzilla/>)

Be sure to mention the document's identifier:

Cluster_Suite_Overview(EN)-5 (2009-08-18T15:49)

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

문서 내용 개선을 위한 제안이 있으실 경우, 최대한 명확히 설명해 주시기 바랍니다. 오류를 발견하셨다면 저희가 쉽게 식별할 수 있도록 섹션 번호와 주위의 문장들을 함께 보내주시기 바랍니다.

1장. RED HAT CLUSTER SUITE 개요

클러스터 시스템은 실제 제품 서비스에 신뢰성, 확장성, 가용성을 제공합니다. Red Hat Cluster Suite를 사용하여 성능, 고가용성, 로드 밸런싱, 확장성, 파일 공유, 경제성에 따라 클러스터를 생성할 수 있습니다. 다음 부분에서는 Red Hat Cluster Suite 구성 요소 및 기능에 대한 개요를 다루고 있으며 이는 다음과 같이 구성되어 있습니다:

- 1.1절. “클러스터 기초”
- 1.2절. “Red Hat Cluster Suite Introduction”
- 1.3절. “Cluster Infrastructure”
- 1.4절. “고가용성 서비스 관리”
- 1.5절. “Red Hat GFS”
- 1.6절. “CLVM (Cluster Logical Volume Manager)”
- 1.7절. “GNBD (Global Network Block Device)”
- 1.8절. “LVS (Linux Virtual Server)”
- 1.9절. “클러스터 관리 도구”
- 1.10절. “Linux 가상 서버 관리 GUI”

1.1. 클러스터 기초

클러스터는 두 개 이상의 컴퓨터가 (노드 또는 *멤버*라고 부름) 작업을 실행하기 위해 함께 동작하는 것으로 이는 네 개의 주요 클러스터 유형으로 되어 있습니다:

- 스토리지 (Storage)
- 고가용성 (High availability)
- 로드 밸런싱 (Load balancing)
- 고성능 (High performance)

스토리지 클러스터 (Storage cluster)는 클러스터에 있는 서버를 통해 일관된 파일 시스템 이미지를 제공하며, 서버의 단일 공유 파일 시스템을 동시에 읽고 쓰는 작업을 할 수 있게 합니다. 스토리지 클러스터는 단순히 하나의 파일 시스템으로 설치를 제한하고 어플리케이션을 패칭하여 관리 사항을 저장합니다. 또한 클러스터 전반에 걸친 파일 시스템으로 어플리케이션 데이터에 대한 중복된 복사본 없이 단순히 백업 및 장애 복구합니다. Red Hat Cluster Suite은 Red Hat GFS를 통해 스토리지 클러스터를 지원합니다.

고가용성 클러스터 (High-availability cluster)는 SPOF (single points of failure)가 없게하고 노드가 실행 불가능하게 될 경우 하나의 클러스터 노드에서 다른 노드로 서비스를 장애 조치하여 지속적으로 서비스를 사용 가능하게 합니다. 일반적으로 고가용성 클러스터에 있는 서비스로 데이터를 읽고 쓰는 작업 (읽기-쓰기 마운트된 파일 시스템을 통해)을 할 수 있습니다. 따라서, 고가용성 클러스터는 하나의 클러스터 노드가 다른 클러스터 노드에서 서비스를 제어함으로써 데이터를 통합적으로 관리하게 합니다. 고가용성 클러스터에서의 노드 장애는 클러스터 외부의 클라이언트에서 확인할 수 없습니다. (고가용성 클러스터는 장애 조치 클러스터라고도 부름) Red Hat Cluster Suite은 고가용성 서비스 관리 구성 요소를 통해 고가용성 클러스터링을 지원합니다.

로드 밸런싱 클러스터 (Load-balancing cluster)는 다중 클러스터 노드로 네트워크 서비스 요청을 내보내어 클러스터 노드 사이의 요청량의 균형을 유지합니다. 로드 밸런싱을 통해 요청량에 따라 노드 수를 맞춤

수 있으므로 확장면에서 비용 효율적입니다. 로드 밸런싱 클러스터에 있는 노드가 실행 불가능하게 될 경우, 로드 밸런싱 소프트웨어는 장애 사항을 감지하여 다른 클러스터 노드에 요청을 보내게 됩니다. 로드 밸런싱 클러스터에서의 노드 장애를 클러스터 외부의 클라이언트가 확인할 수 없습니다. Red Hat Cluster Suite는 LVS (Linux Virtual Server)를 통해 로드 밸런싱을 지원합니다.

고성능 클러스터 (High-performance cluster)는 클러스터 노드를 사용하여 동시 연산을 실행합니다. 고성능 클러스터로 애플리케이션이 병렬적 작업을 할 수 있게 되어 애플리케이션의 성능을 향상시킬 수 있습니다. (고성능 클러스터는 컴퓨팅 클러스터 또는 그리드 컴퓨팅이라고 합니다.)



참고

이전 부분의 클러스터 유형에 관한 요약에서는 기본적인 설정 사항을 다루고 있습니다; 필요에 따라 클러스터 기능을 통합적으로 사용할 수도 있습니다.

1.2. RED HAT CLUSTER SUITE INTRODUCTION

Red Hat Cluster Suite (RHCS)은 통합된 소프트웨어 구성 요소 모음으로 다양한 설정 사항에서 성능, 고가용성, 로드 밸런싱, 확장성, 파일 공유성, 경제성과 같이 필요에 따라 사용될 수 있습니다.

RHCS consists of the following major components (refer to [그림 1.1. “Red Hat Cluster Suite Introduction”](#)):

- 클러스터 인프라 – 클러스터로 함께 동작하기 위해 노드에 기본적인 기능을 제공합니다. 이에는 설정-파일 관리, 멤버십 관리, 잠금 관리, **fencing**과 같은 기능이 있습니다.
- 고가용성 서비스 관리 – 노드가 작동하지 않을 경우 하나의 클러스터 노드에서 다른 노드로 장애 조치 서비스를 제공합니다.
- 클러스터 관리 도구 – Red Hat 클러스터 셋업, 설정, 관리에 필요한 설정 및 관리 도구입니다. 클러스터 인프라 구성 요소, 고가용성, 서비스 관리 구성요소, 스토리지와 함께 사용되는 도구입니다.
- LVS (Linux Virtual Server) – IP 로드 밸런싱을 제공하는 라우팅 소프트웨어입니다. LVS 서버 뒤에 있는 실제 서버로 클라이언트 요청을 균등하게 보내는 중복서버 쌍에서 LVS가 실행됩니다.

(Red Hat Cluster Suite의 일부분이 *아닌*) 패키지 옵션의 일부분인 다음과 같은 구성 요소를 사용하여 Red Hat Cluster Suite을 보완할 수 있습니다.

- Red Hat GFS (Global File System) – Red Hat Cluster Suite과 함께 사용할 클러스터 파일 시스템을 제공합니다. 스토리지가 각각의 클러스터 노드에 로컬로 연결되어 있는 것처럼 GFS는 다중 노드가 블록 레벨에서 스토리지를 공유하게 합니다.
- CLVM (Cluster Logical Volume Manager) – 클러스터 스토리지의 볼륨을 관리합니다.



참고

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 [1.6절. “CLVM \(Cluster Logical Volume Manager\)”](#).

- GNBD (Global Network Block Device) – GFS의 보조 구성 요소로 이더넷에 블록 레벨 스토리지를 내보냅니다. 이는 블록 레벨 스토리지를 Red Hat GFS에서 사용 가능하게 하기 위한 경제적인 방법입니다.

For a lower level summary of Red Hat Cluster Suite components and optional software, refer to [2장. Red Hat Cluster Suite 구성 요소 요약](#)

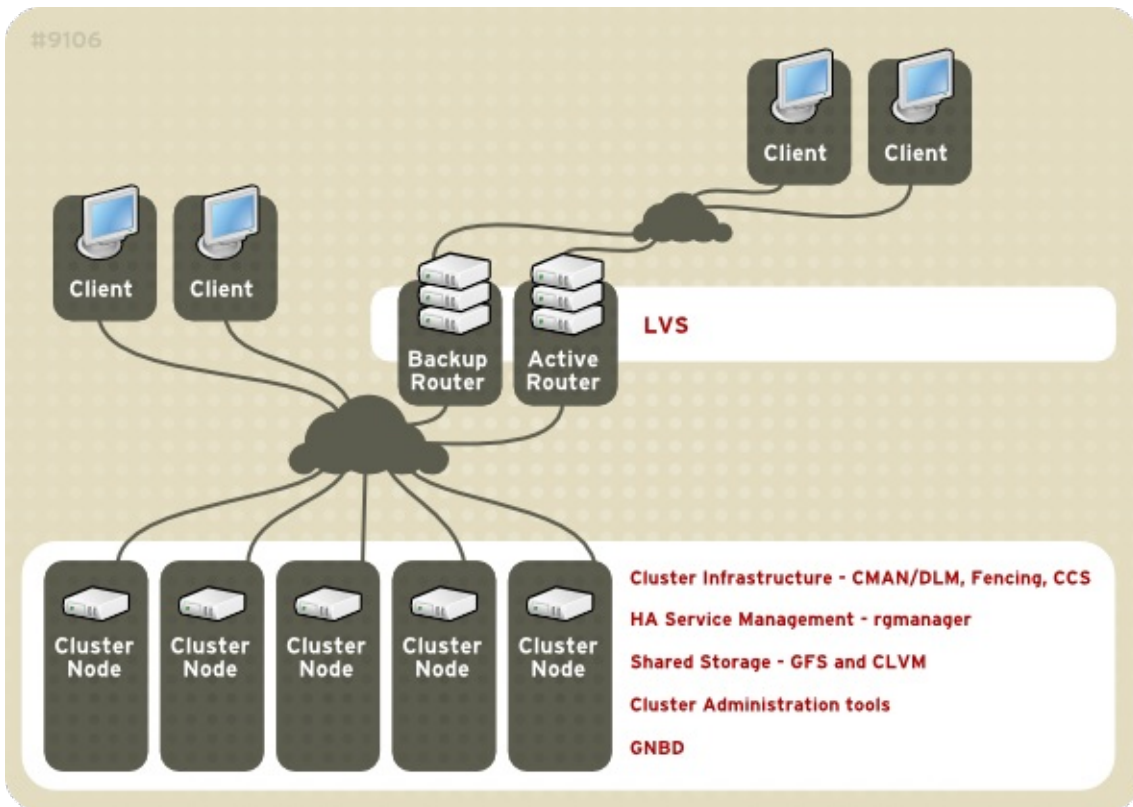


그림 1.1. Red Hat Cluster Suite Introduction



참고

그림 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

Red Hat Cluster Suite 클러스터 인프라는 클러스터로 함께 작동하여 컴퓨터 그룹 (노드 또는 *멤버*라고 부름)에 필요한 기본적인 기능을 제공합니다. 클러스터 인프라를 사용하여 클러스터가 구성되면, 사용자의 클러스터링 요구에 따라 기타 다른 Red Hat Cluster Suite 구성 요소를 사용할 수 있습니다. (예, GFS 파일 시스템에서 공유 파일에 필요한 클러스터 셋업 또는 서비스 장애 조치 셋업) 클러스터 인프라로 다음과 같은 기능을 실행할 수 있습니다:

- 클러스터 관리
- 잠금 관리
- Fencing
- 클러스터 설정 관리

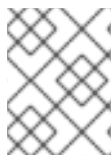
1.3.1. 클러스터 관리

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 그림 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.

정족수 (quorum)는 이더넷을 통한 클러스터 노드 사이에서의 메시지 통신에 의해 결정됩니다. 다른 방법으로 정족수는 이더넷을 통한 메시지 통신과 정족수 디스크를 통한 메시지 통신의 결합으로 결정될 수 있습니다. 이더넷을 통한 정족수의 경우, 정족수는 플러스 1을 선택한 노드의 50 퍼센트로 구성되며, 정족수 디스크를 통한 정족수의 경우, 정족수는 사용자 지정 상태로 구성됩니다.



참고

기본값으로 각각의 노드는 하나의 정족수 투표(quorum vote)를 갖습니다. 다른 방법으로 각각의 노드가 하나 이상의 정족수 투표를 갖도록 설정하실 수 있습니다.

CMAN은 다른 클러스터 노드에서의 메시지를 모니터링하여 멤버십을 파악합니다. 클러스터 멤버십이 변경될 때 클러스터 관리자는 다른 인프라 구성 요소를 확인하고 적절한 작업을 실행합니다. 예를 들어, 노드 A가 클러스터에 연결되어 노드 B와 C가 마운트된 GFS 파일 시스템을 마운트할 경우, GFS 파일 시스템을 이용하기 위해 노드 A에 추가 저널 및 잠금 관리 기능을 필요로 합니다. 클러스터 노드가 지정된 시간 안에 메시지를 전송하지 않을 경우, 클러스터 관리자는 클러스터에서 노드를 삭제하고 멤버가 아닌 노드가 있는 다른 클러스터 인프라 구성 요소와 통신합니다. 다른 클러스터 인프라 구성 요소는 더이상 클러스터 멤버가 아닌 노드에 대해 어떤 작업을 실행할 지를 결정합니다. 예를 들어, Fencing으로 멤버가 아닌 노드를 차단할 수 있습니다.

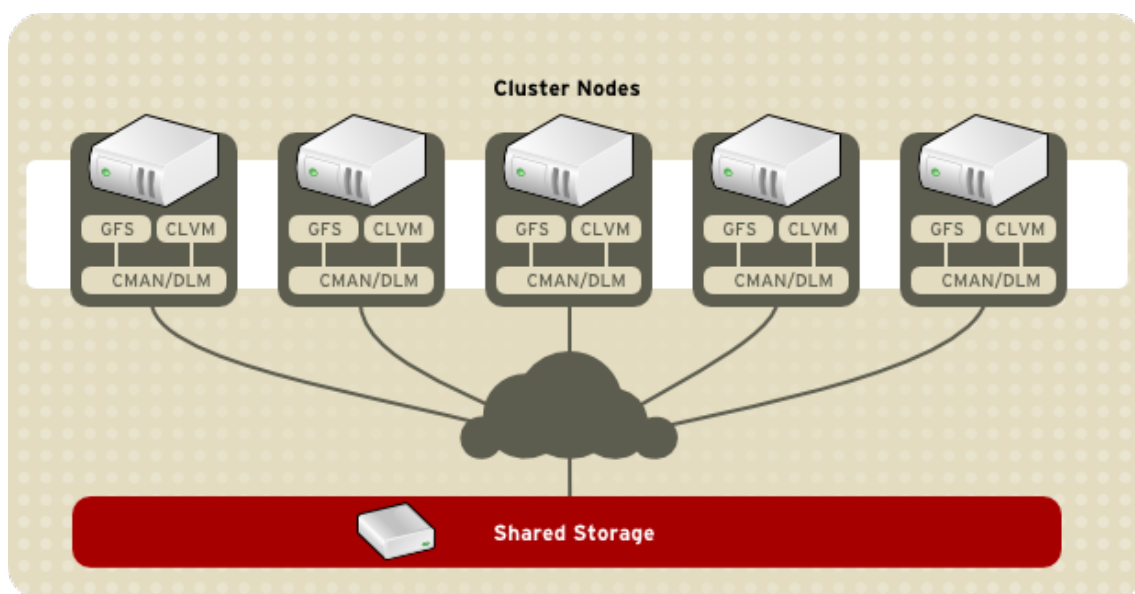


그림 1.2. CMAN/DLM Overview

1.3.2. 잠금 관리

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 [그림 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**.

CMAN이 노드에 장애가 발생했다고 판단하면, 장애가 발생한 노드가 있는 다른 클러스터 인프라 구성 요소와 통신합니다. 장애 발생 통보시 **fenced**는 장애 발생 노드를 차단합니다. 다른 클러스터 인프라 구성 요소는 어떤 작업을 취할 지를 결정합니다 – 즉, 노드에 필요한 복구 작업을 실행합니다. 예를 들어, 노드 장애 발생시 **fenced**가 장애가 발생한 노드에 **fencing** 작업을 완료할 때까지 **DLM** 및 **GFS**는 작업을 일시 중지합니다. 장애 발생한 노드의 **fencing** 작업이 완료되면, **DLM** 및 **GFS**는 복구 작업을 실행합니다. **DLM**은 장애 발생 노드의 잠금 기능을 해제하고, **GFS**는 장애 발생 노드의 저널을 복구합니다.

fencing 프로그램은 클러스터 설정 파일에서 어떤 **fencing** 방법을 사용할 지를 결정합니다. 클러스터 설정 파일에서는 **fencing** 에이전트 및 **fencing** 장치라는 두가지 주요 **fencing** 방법이 있습니다: **fencing** 프로그램은 클러스터 설정 파일에 지정된 **fencing** 에이전트로 호출합니다. 다음으로 **fencing** 에이전트는 **fencing** 장치를 통해 노드를 차단합니다. **fencing** 작업이 완료되면 **fencing** 프로그램은 클러스터 관리자에게 알립니다.

Red Hat Cluster Suite에서는 다양한 **fencing** 방법을 제공합니다:

- **Power fencing** – 실행 불가능한 노드의 전원을 끄기위해 전원 제어를 사용하는 **fencing** 방법
- **Fibre Channel switch fencing** – 스토리지를 실행 불가능한 노드로 연결시키는 Fibre Channel 포트를 비활성화시키는 **fencing** 방법
- **GNBD fencing** – A fencing method that disables an inoperable node's access to a GNBD server.
- 기타 **fencing** 방법 – IBM Bladecenters, PAP, DRAC/MC, HP ILO, IPMI, IBM RSA II, 기타 등등을 포함하여 실행 불가능한 노드의 I/O 또는 전원을 비활성화시키는 기타 다른 **fencing** 방법

그림 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. 그림 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.

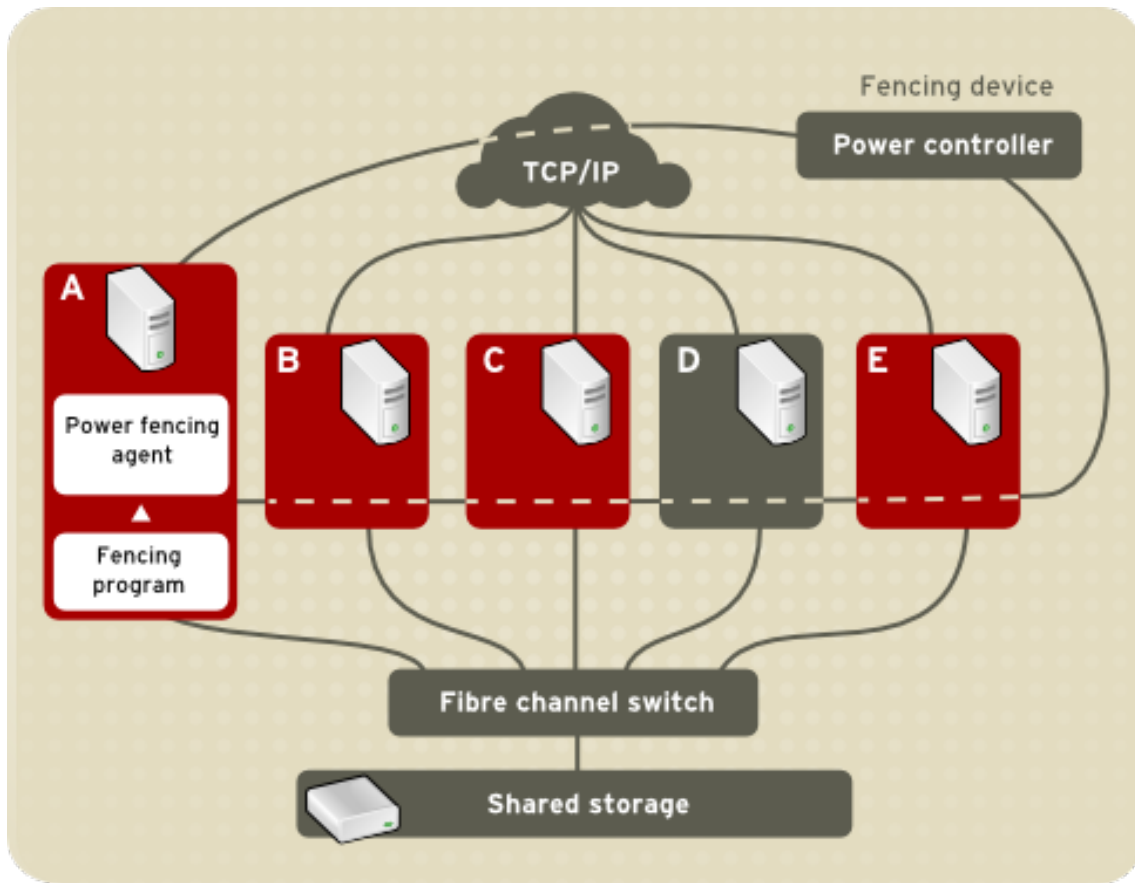


그림 1.3. Power Fencing Example

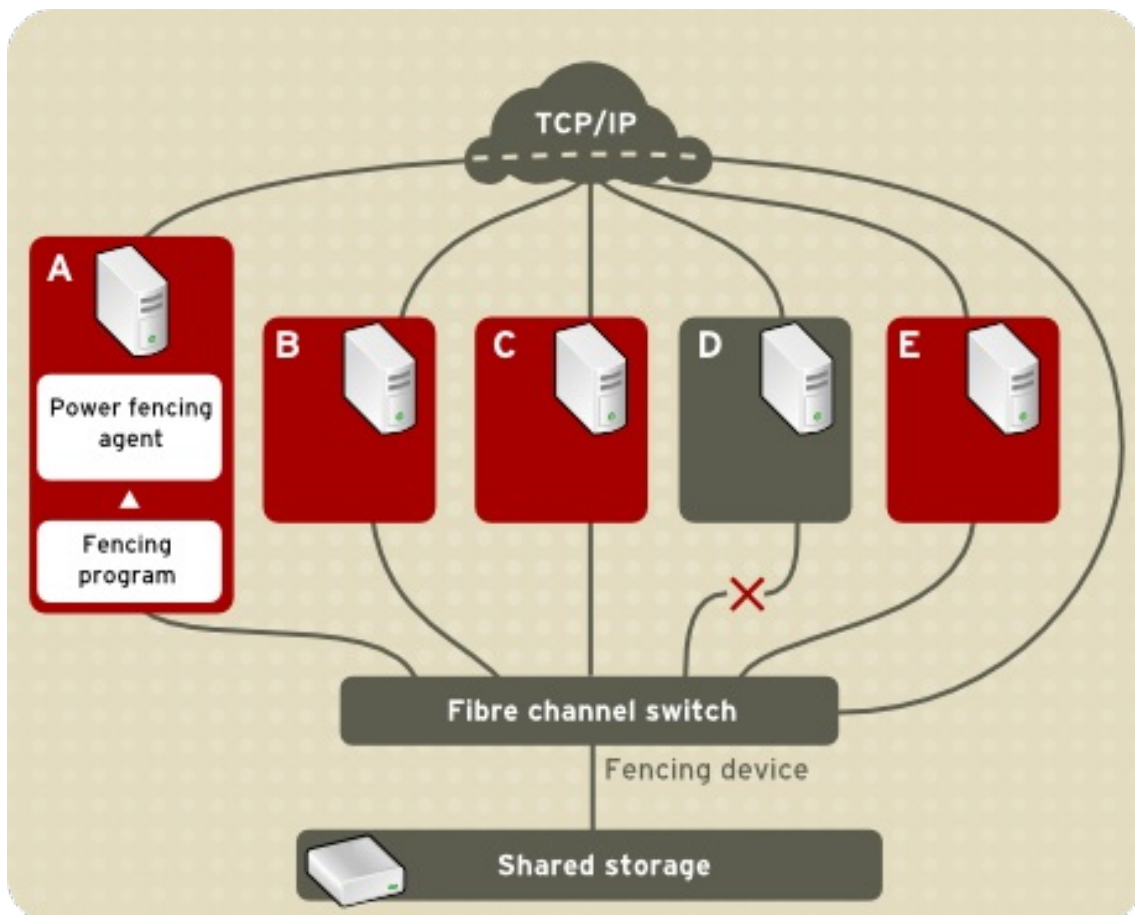


그림 1.4. Fibre Channel Switch Fencing Example

fencing 방법은 클러스터에 있는 각각의 노드에 대한 fencing 방법 이름, fencing 에이전트, fencing 장치를 지정하기 위한 클러스터 설정 파일의 편집으로 구성되어 있습니다.

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 [그림 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 [그림 1.6. “Fencing a Node with Dual Fibre Channel Connections”](#)).

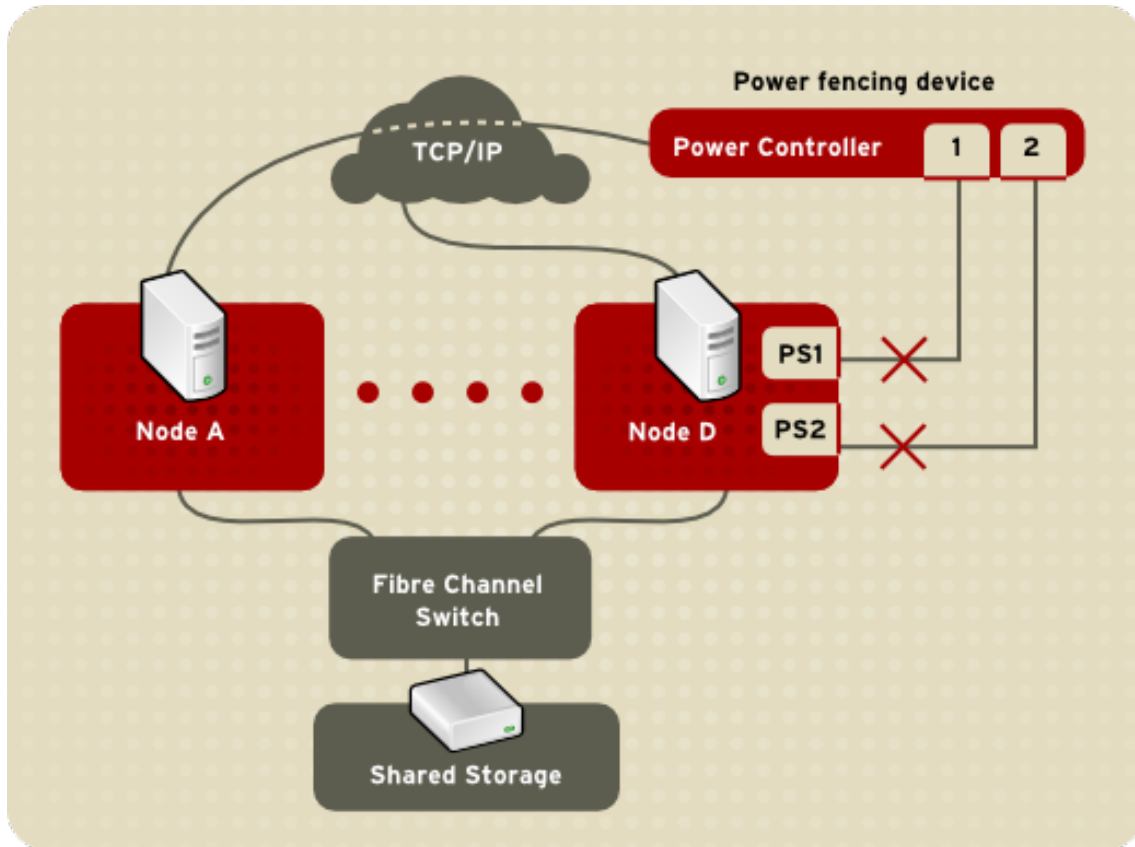


그림 1.5. Fencing a Node with Dual Power Supplies

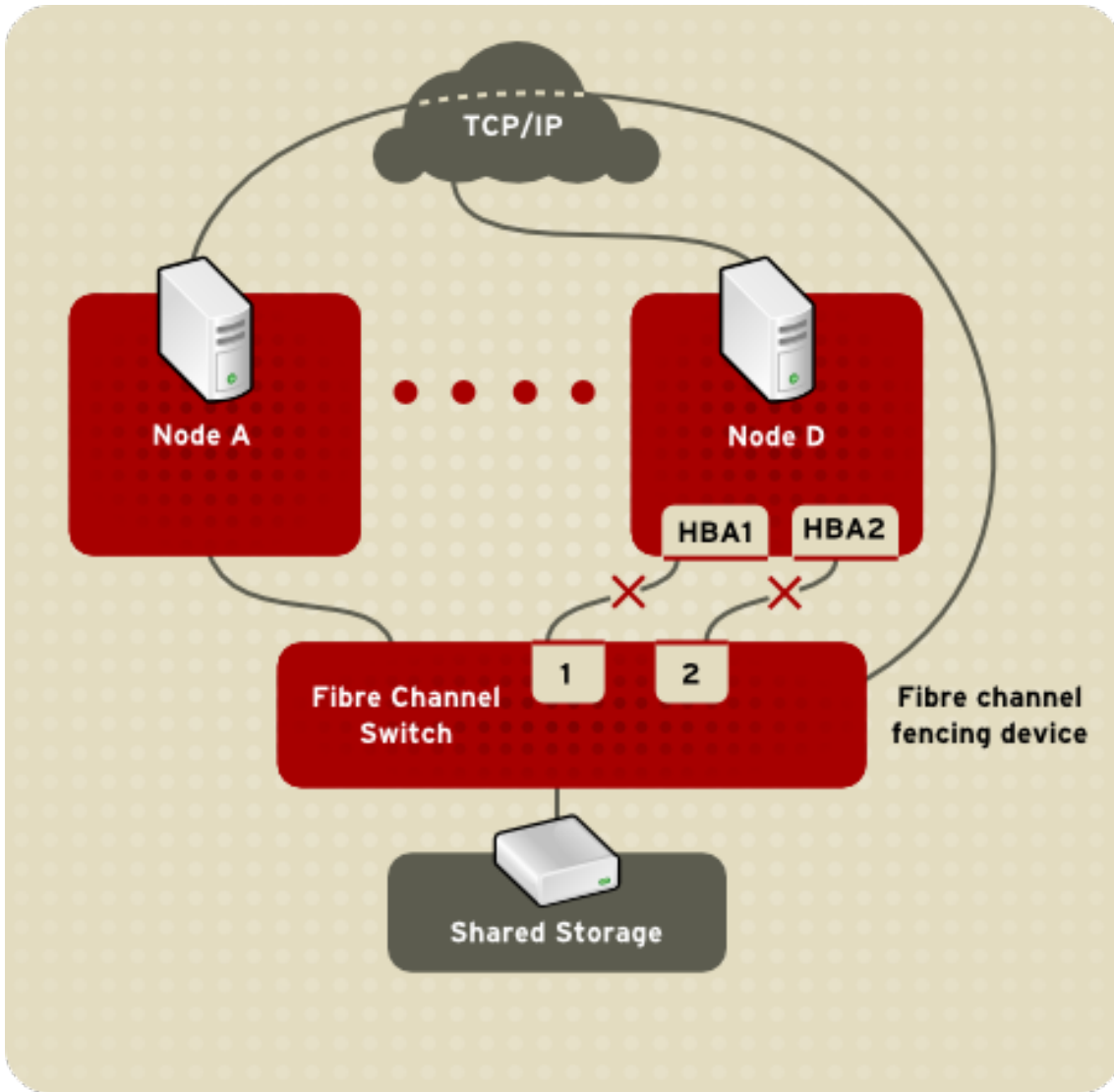


그림 1.6. Fencing a Node with Dual Fibre Channel Connections

하나의 fencing 방법이나 또는 여러 개의 fencing 방법을 사용하여 노드를 설정할 수 있습니다. 하나의 fencing 방법으로 노드를 설정한 경우, 이 방법만으로 노드를 fencing할 수 있습니다. 여러 fencing 방법으로 노드를 설정한 경우, 클러스터 설정 파일에서 지정된 fencing 방법 순서에 따라 하나의 fencing 방법에서 다른 방법으로 *캐스캐이드 (cascade)* 됩니다. 노드에 문제가 있을 경우, 클러스터 설정 파일에서 설정한 첫번째 fencing 방법으로 노드가 차단됩니다. 첫번째 fencing 방법에 문제가 발생했을 경우 두번째로 설정된 fencing 방법이 사용됩니다. 아무 fencing 방법도 실행되지 않을 경우, fencing은 첫번째로 설정된 fencing 방법을 다시 실행하게 되며 노드에 fencing이 실행될 때까지 클러스터 설정 파일에 설정된 순서로 fencing 방법을 반복 실행하게 됩니다.

1.3.4. 클러스터 설정 시스템

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 [그림 1.7. “CCS Overview”](#)).

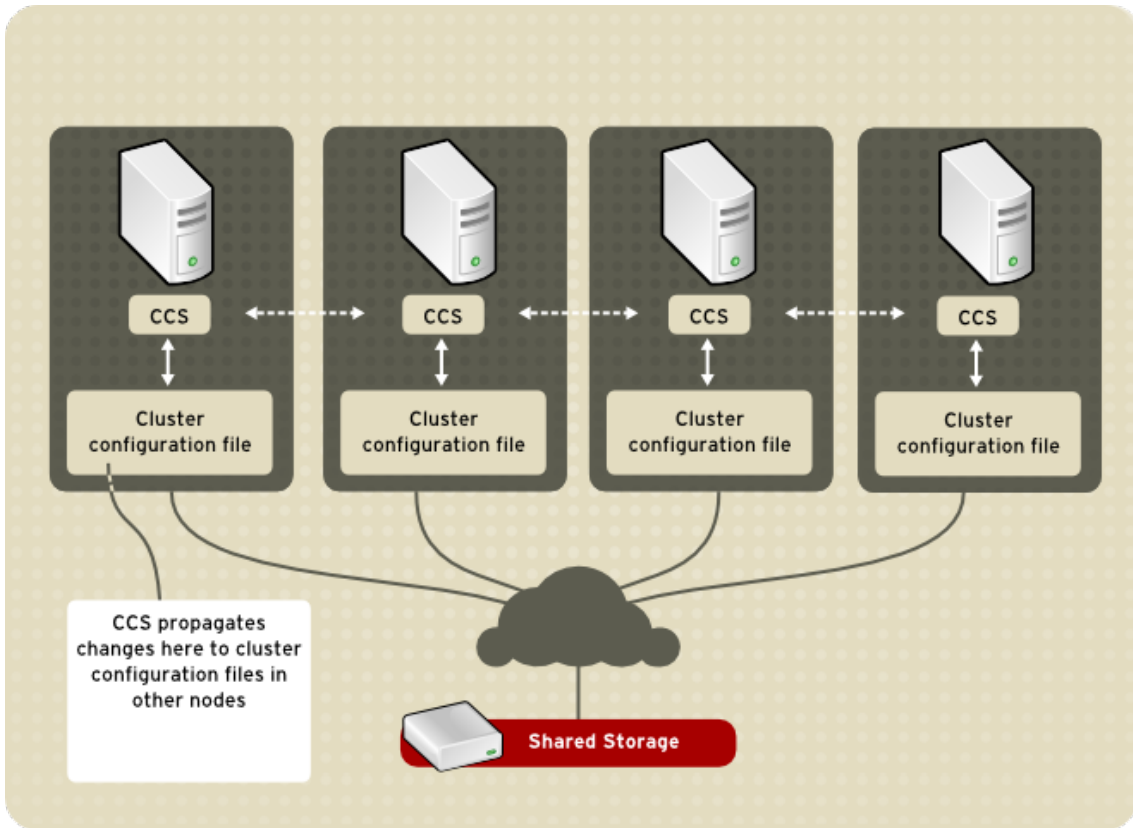


그림 1.7. CCS Overview

Other cluster components (for example, CMAN) access configuration information from the configuration file through CCS (refer to [그림 1.7. “CCS Overview”](#)).

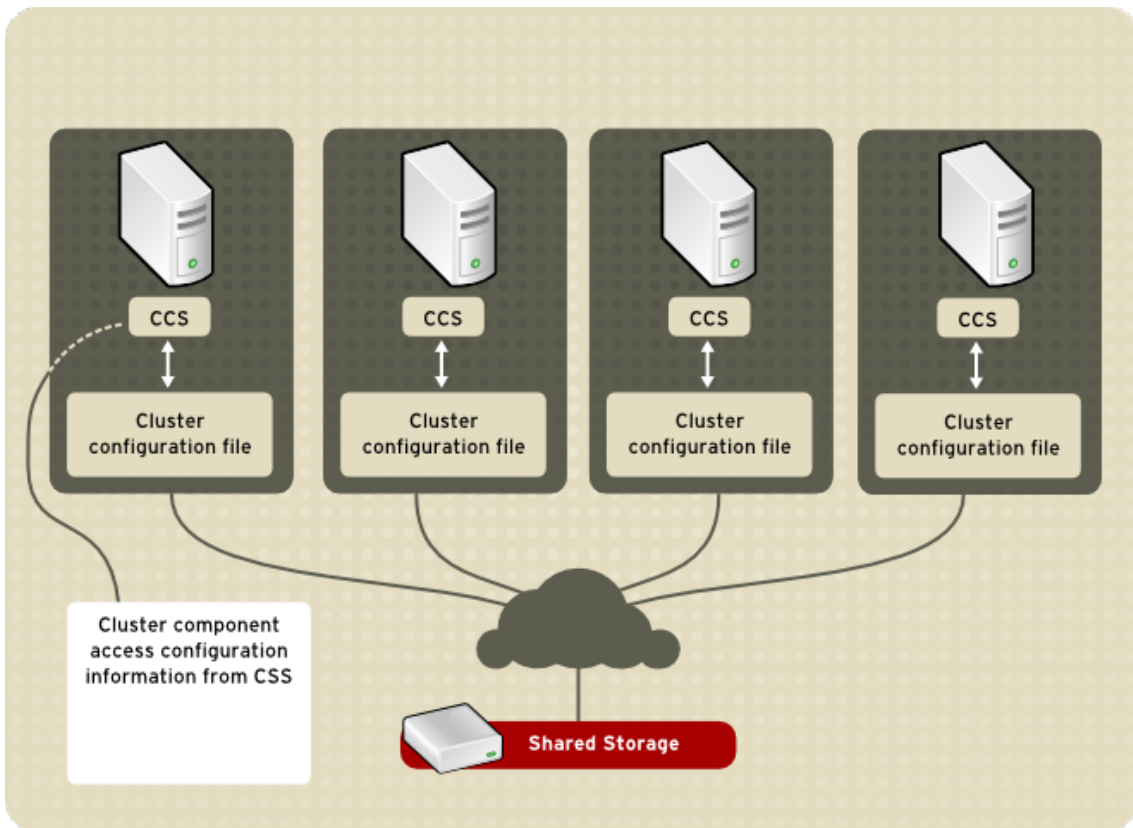


그림 1.8. Accessing Configuration Information

클러스터 설정 파일 (`/etc/cluster/cluster.conf`)은 XML 파일로 다음과 같은 클러스터 특징이 설명되어 있습니다:

- 클러스터 이름 – 클러스터 이름, 클러스터 설정 파일 개정 레벨, 노드가 클러스터에 결합할 때에나 클러스터에서 **fence**될 때 사용되는 기본적인 **fence** 타이밍 등록 정보를 보여 줍니다.
- 클러스터 – 클러스터의 각 노드, 설정된 노드 이름, 노드 ID, 정족수 투표수, 노드의 **fencing** 방법을 보여줍니다.
- Fence 장치 – 클러스터에 있는 **fence** 장치를 보여줍니다. 매개 변수는 **fence** 장치의 유형에 따라 다양합니다. 예를 들어 **fence** 장치로 전원 제어가 사용될 경우, 클러스터 설정에서는 전원 제어기의 이름, 제어기의 IP 주소, 로그인, 암호를 지정합니다.
- 관리 리소스 – 클러스터 서비스를 생성하는데 필요한 리소스를 보여줍니다. 관리 리소스에는 장애 조치 도메인에 대한 정의, 리소스 (예: IP 주소), 서비스가 포함됩니다. 관리 리소스에서는 클러스터 서비스와 클러스터 서비스의 장애 조치 기능 모두를 지정합니다.

1.4. 고가용성 서비스 관리

고가용성 서비스 관리에서는 Red Hat 클러스터에 있는 고가용성 *클러스터 서비스*를 생성하고 관리하는 기능을 제공합니다. Red Hat 클러스터에 있는 고가용성 서비스 관리에 필요한 주요 구성요소인 **rgmanager**는 상용 어플리케이션에 필요한 **cold failover**를 실행합니다. Red Hat 클러스터에서 어플리케이션은 고가용성 클러스터 서비스를 구축하기 위해 다른 클러스터 리소스로 설정되어 있습니다. 고가용성 클러스터 서비스는 클러스터 클라이언트에 장애를 일으키지 않고 하나의 클러스터 노드에서 다른 노드로 장애 조치될 수 있습니다. 클러스터 노드에 문제가 발생한 경우나 또는 클러스터 시스템 관리자가 다른 클러스터 노드로 서비스를 이전할 경우 클러스터 서비스 장애 조치가 실행될 수 있습니다. (예: 클러스터 노드의 정기적 중지와 같은 경우)

고가용성 서비스를 생성하려면 이를 클러스터 설정 파일에서 설정해야 합니다. 클러스터 서비스는 클러스터 리소스로 구성되어 있습니다. 클러스터 리소스에서는 클러스터 설정 파일에서 생성하고 관리할 블록을 만듭니다. – 예: IP 주소, 어플리케이션 초기화 스크립트, RHGFS; 공유 파티션

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 [그림 1.9. “장애조치 도메인”](#)).



참고

장애조치 도메인은 작동에 필요하지 *않습니다*.

클러스터 서비스는 데이터 무결성을 유지하기 위해 한번에 하나의 클러스터 노드에서만 실행될 수 있습니다. 장애조치 도메인에서 장애 조치 우선 순위를 지정할 수 있습니다. 지정된 장애 조치 우선 순위는 장애 조치 도메인에 있는 각각의 노드에 지정된 우선 순위로 구성되어 있습니다. 우선 순위로 장애 조치 순서가 결정됩니다 – 클러스터 서비스가 어떤 노드를 장애 조치할 지를 결정. 장애 조치 우선 순위를 지정하지 않으셨을 경우, 클러스터 서비스는 장애 조치 도메인에 있는 아무 노드에 장애 조치를 실행할 수 있습니다. 또한, 클러스터 서비스를 관련 장애 조치 도메인의 노드에서만 실행하도록 제한할 수 있습니다. (제한되지 않은 장애 조치 도메인과 연관될 경우, 사용 가능한 장애 조치 도메인이 없을 경우 클러스터 서비스는 아무 클러스터 노드에서 시작될 수 있습니다.)

In [그림 1.9. “장애조치 도메인”](#), 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.

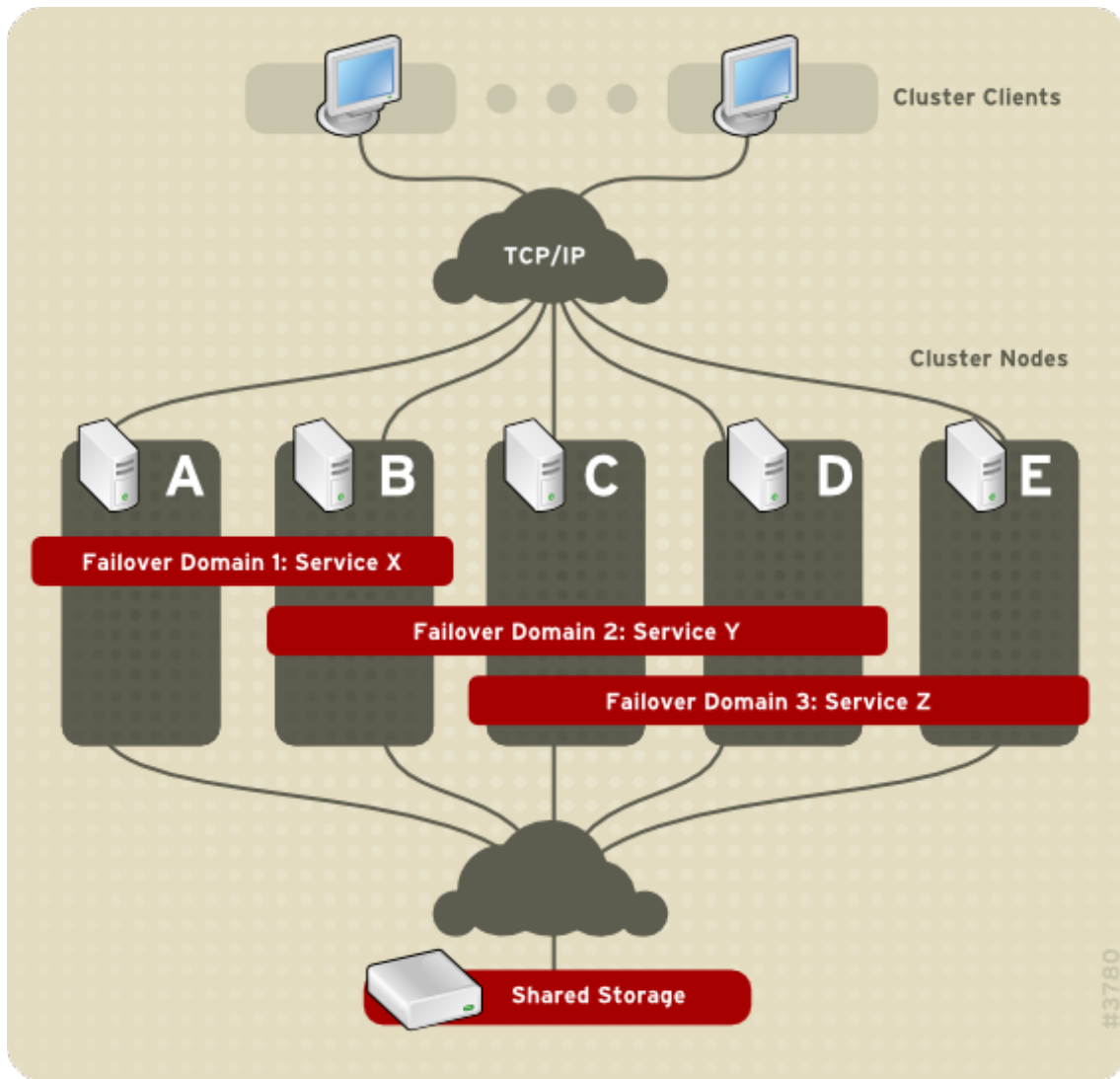


그림 1.9. 장애조치 도메인

그림 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:

- IP 주소 리소스 – IP 주소 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".

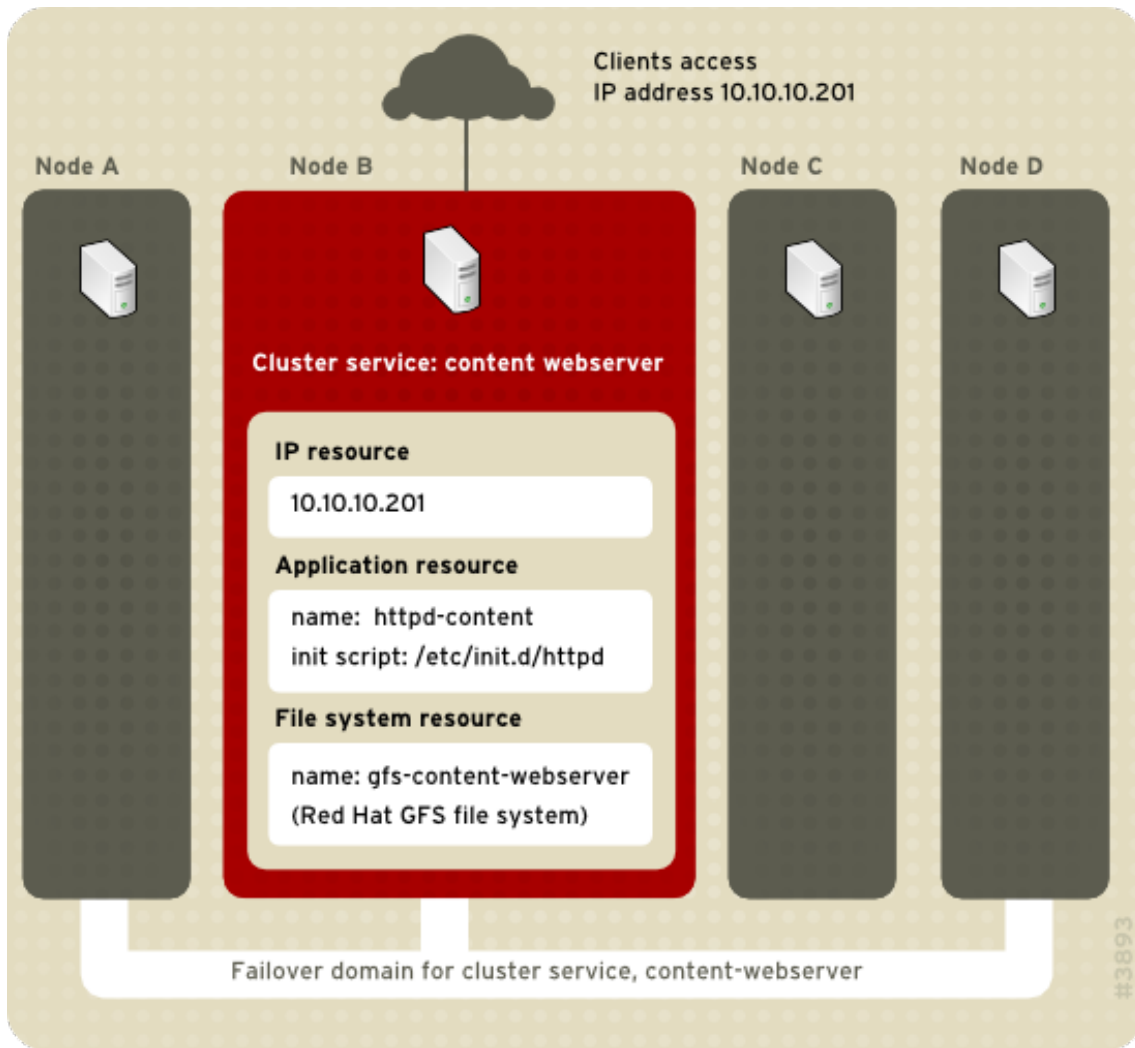


그림 1.10. Web Server Cluster Service Example

클라이언트는 10.10.10.201 IP 주소를 통해 클러스터 서비스에 액세스하며, 웹서버 어플리케이션인 `httpd-content`와 상호 대화합니다. `httpd-content` 어플리케이션은 `gfs-content-webserver` 파일 시스템을 사용합니다. 노드 B에 장애가 발생할 경우, `content-webserver` 클러스터 서비스는 노드 D로 장애 조치를 실행합니다. 노드 D가 사용 불가능하거나 또는 장애가 발생할 경우, 노드 A로 장애조치합니다. 클러스터 클라이언트에 명백한 장애가 없을 경우 장애 조치가 실행됩니다. 클러스터 서비스는 장애 조치 되기 전과 같은 IP 주소를 통해 다른 클러스터 노드에서 액세스될 수 있습니다.

1.5. RED HAT GFS

Red Hat GFS는 클러스터 노드가 노드 사이에서 공유되는 블록 장치로 동시에 액세스하게 하는 클러스터 파일 시스템입니다. GFS는 Linux 커널 파일 시스템 인터페이스의 VFS 레이어와 직접적으로 통신하는 파일 시스템입니다. GFS는 클러스터에서의 실행 최적화를 위해 분산 메타데이터와 다중 저널을 사용합니다. 파일 시스템 무결성을 유지하기 위해 GFS는 I/O를 관리하여 잠금 관리 기능을 사용합니다. 하나의 노드가 GFS 파일 시스템에서 데이터를 변경할 경우, 파일 시스템을 사용하여 다른 클러스터 노드에서 변경 사항을 직접 확인할 수 있습니다.

Red Hat GFS를 사용하면 다음과 같은 장점을 통해 최대로 어플리케이션 업타임을 극대화할 수 있습니다:

- 데이터 기반 구조의 단순화
 - 모든 클러스터에 대해 한 번만 어플리케이션을 설치 및 패치
 - 어플리케이션 데이터의 여유 사본(복사본)의 필요성 배제

- 여러 클라이언트가 동시에 데이터로 읽기/쓰기 작업 액세스 가능
- 백업 및 장애 복구의 단순화 (하나의 파일 시스템에만 백업 또는 복구)
- 스토리지 리소스 사용을 극대화; 스토리지 관리 비용을 최소화
 - 파티션 대신 전체적으로 스토리지 관리
 - 데이터 중복을 감소시켜 전체 스토리지의 니즈를 최소화
- 클러스터를 동일하게 확장시키고 스토리지나 서버를 즉시 추가
 - 복잡한 테크닉 없이 스토리지 파티션하기
 - 동일 파일 시스템에 서버를 마운팅하여 간편하게 서버 추가

Red Hat GFS을 실행하는 노드는 Red Hat Cluster Suite 설정 및 관리 도구로 설정 및 관리됩니다. 볼륨 관리는 CLVM (Cluster Logical Volume Manager)을 통해 관리됩니다. Red Hat GFS는 Red Hat 클러스터에 있는 GFS 노드의 데이터를 공유합니다. GFS는 Red Hat 클러스터에 있는 GFS 노드를 통해 파일 시스템 이름 공간을 일관적으로 확인할 수 있습니다. GFS는 어플리케이션이 스토리지 인프라의 기본적인 지식 없이 설치 및 실행하게 합니다. 또한 GFS는 쿼터, 다중 저널, 다중 경로 지원과 같이 일반적으로 엔터프라이즈 환경에 필요한 기능을 제공합니다.

GFS는 스토리지 환경에 필요한 성능, 확장성, 경제성에 따라 네트워킹 스토리지의 다양한 방법을 제공합니다. 다음 부분에서는 GFS에 대한 이해를 돕기 위해 이에 대해 간략하고 기본적인 내용을 다루고 있습니다.

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 [1.7절. “GNBD \(Global Network Block Device\)”](#).)

다음 부분에서는 성능, 확장성, 경제성에 맞도록 GFS를 사용하는 방법에 관한 예를 다루고 있습니다:

- [1.5.1절. “우수한 성능 및 확장성”](#)
- [1.5.2절. “성능, 확장성, 적절한 가격”](#)
- [1.5.3절. “경제성 및 성능”](#)



참고

GFS 사용 예에서는 기본 설정 사항을 다루고 있습니다; 예에서 볼 수 있듯이 클러스터 설정 사항을 통합적으로 사용할 수도 있습니다.

1.5.1. 우수한 성능 및 확장성

You can obtain the highest shared-file performance when applications access storage directly. The GFS SAN configuration in [그림 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.

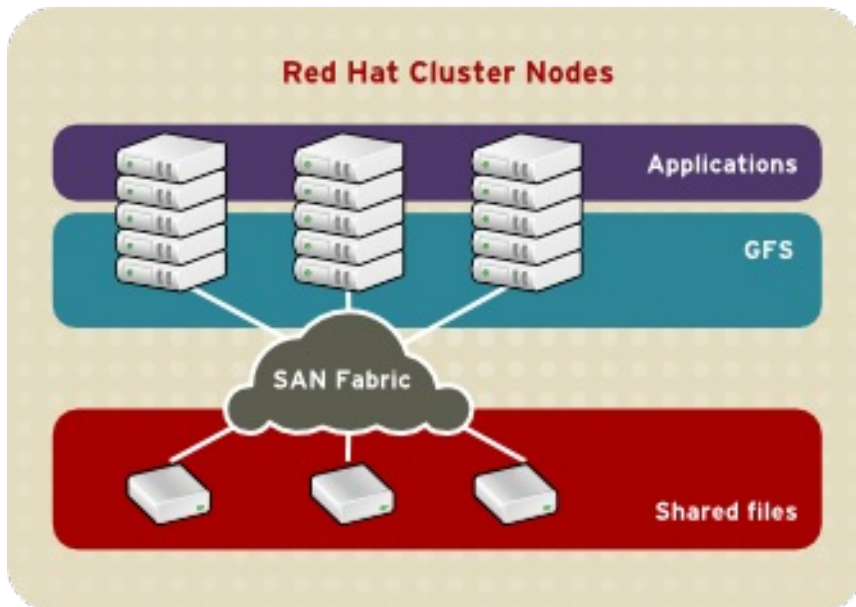


그림 1.11. GFS with a SAN

1.5.2. 성능, 확장성, 적절한 가격

Multiple Linux client applications on a LAN can share the same SAN-based data as shown in [그림 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.

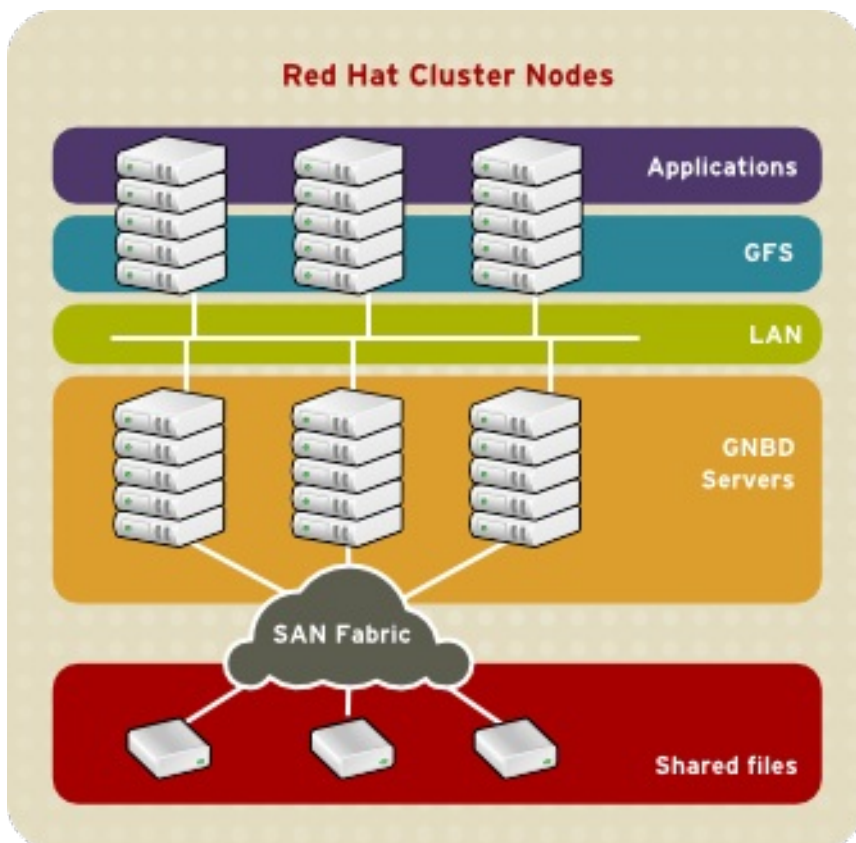


그림 1.12. GFS and GNBD with a SAN

1.5.3. 경제성 및 성능

그림 1.13. “직접 연결된 스토리지를 사용하는 GFS 및 GNBD” 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.

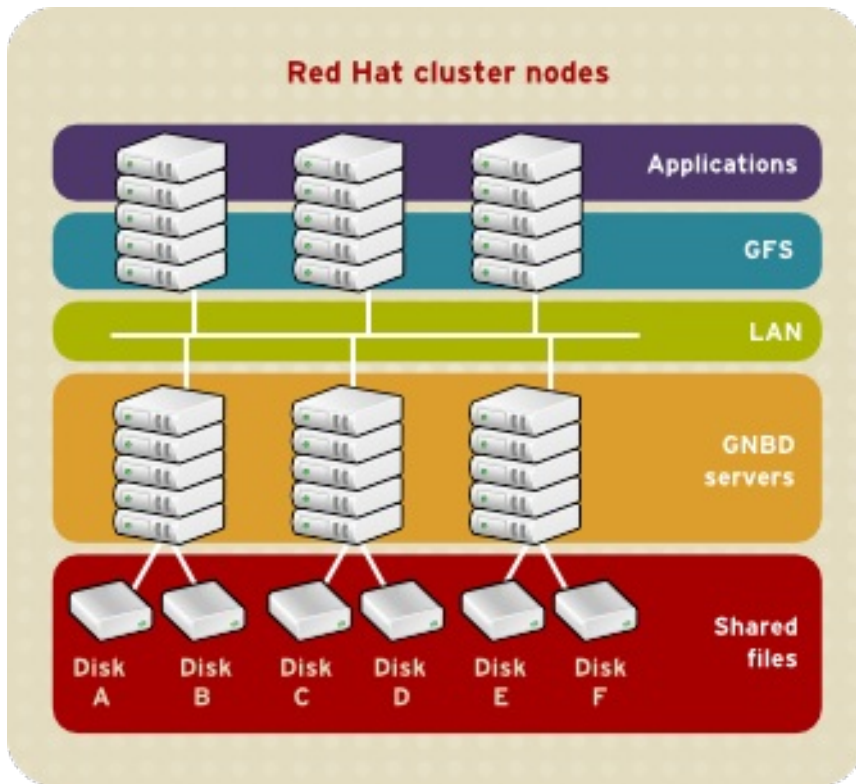


그림 1.13. 직접 연결된 스토리지를 사용하는 GFS 및 GNBD

1.6. CLVM (CLUSTER LOGICAL VOLUME MANAGER)

CLVM (Cluster Logical Volume Manager)은 클러스터 전반에 걸쳐 LVM2 버전을 제공합니다. CLVM은 단일 노드 상의 LVM2와 같은 기능을 제공하지만 Red Hat 클러스터에서는 모든 노드에서 논리 볼륨을 사용할 수 있습니다. CLVM으로 생성된 논리 볼륨은 클러스터에 있는 모든 노드에서 논리 볼륨을 사용 가능하게 합니다.

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 [그림 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 [1.3절. “Cluster Infrastructure”](#)).



참고

Red Hat Cluster Suite에서 공유 스토리지를 사용하려면 클러스터 논리 볼륨 관리자 데몬 (**clvmd**)이나 HA-LVM (High Availability Logical Volume Management) 에이전트를 실행하고 있어야 합니다. **clvmd** 데몬이나 HA-LVM을 실행상의 이유나 해당 인타이틀먼트를 갖고 있지 않아 사용할 수 없을 경우, 데이터 손실을 가져올 수 있으므로 공유 디스크상에서 단일 인스턴스 LVM을 사용해서는 안됩니다. 다른 문제가 있을 경우 Red Hat 서비스 담당자에게 연락하시기 바랍니다.



참고

CLVM 사용을 위해 `/etc/lvm/lvm.conf`에 클러스터 전반에 걸친 잠금 기능에 필요한 사항을 변경해야 합니다.

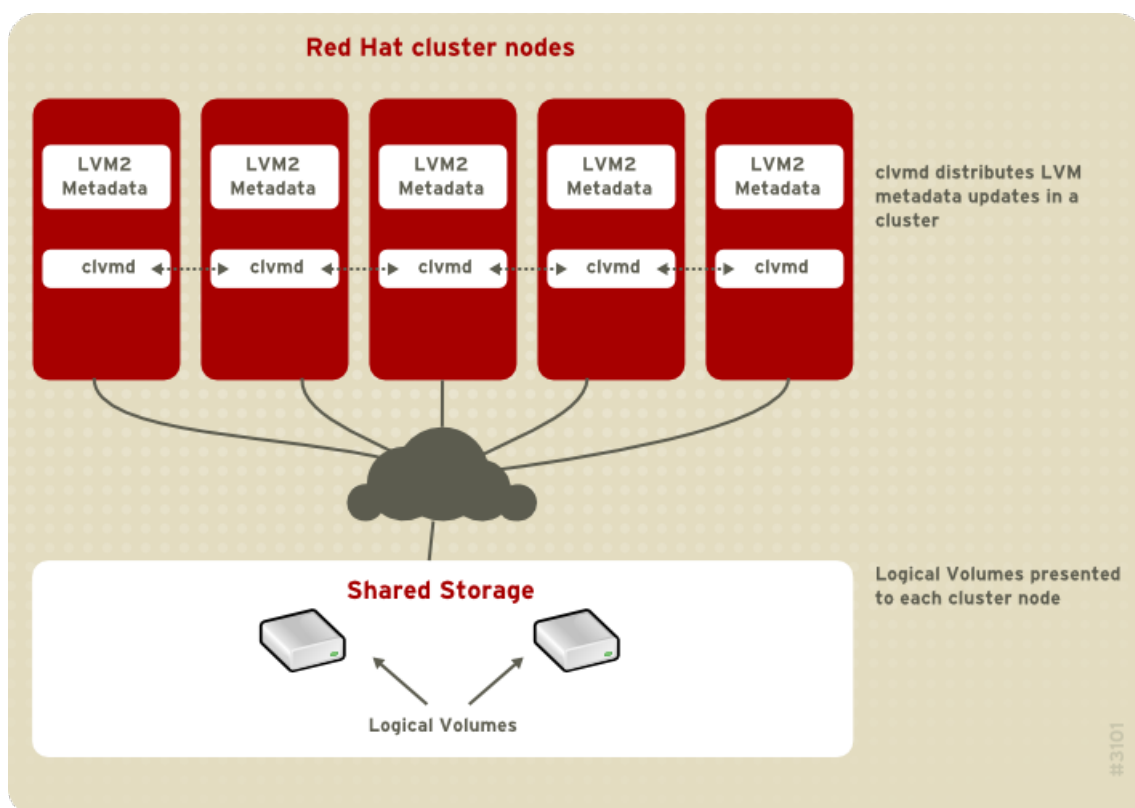


그림 1.14. CLVM Overview

You can configure CLVM using the same commands as LVM2, using the LVM graphical user interface (refer to [그림 1.15. “LVM Graphical User Interface”](#)), or using the storage configuration function of the Conga cluster configuration graphical user interface (refer to [그림 1.16. “Conga LVM Graphical User Interface”](#)). [그림 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.

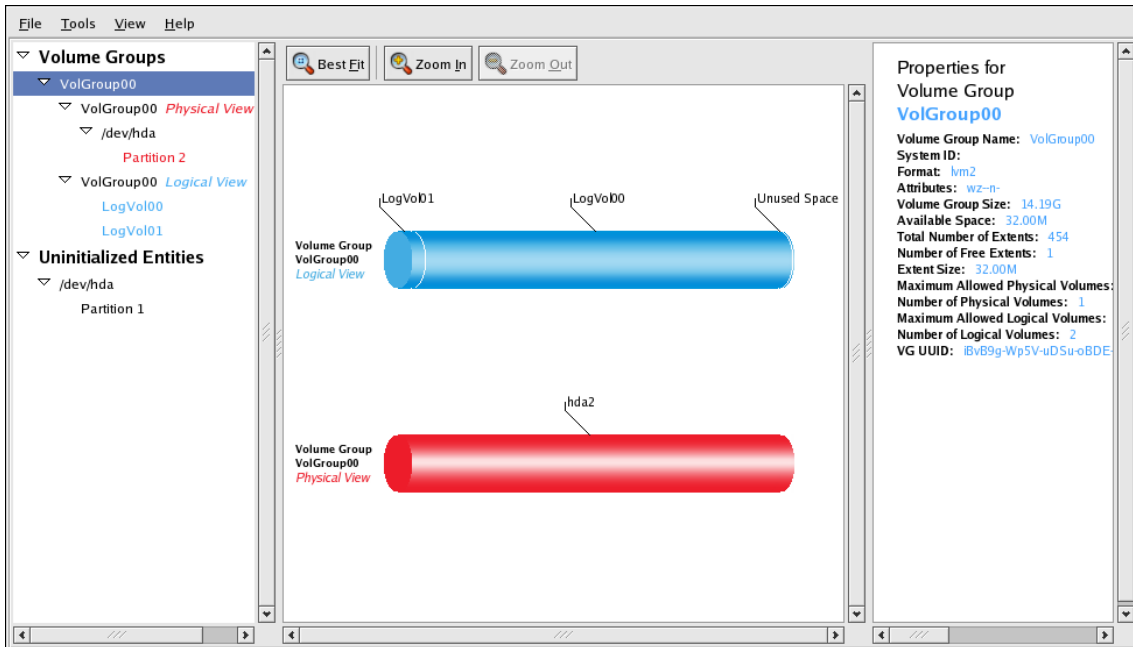


그림 1.15. LVM Graphical User Interface

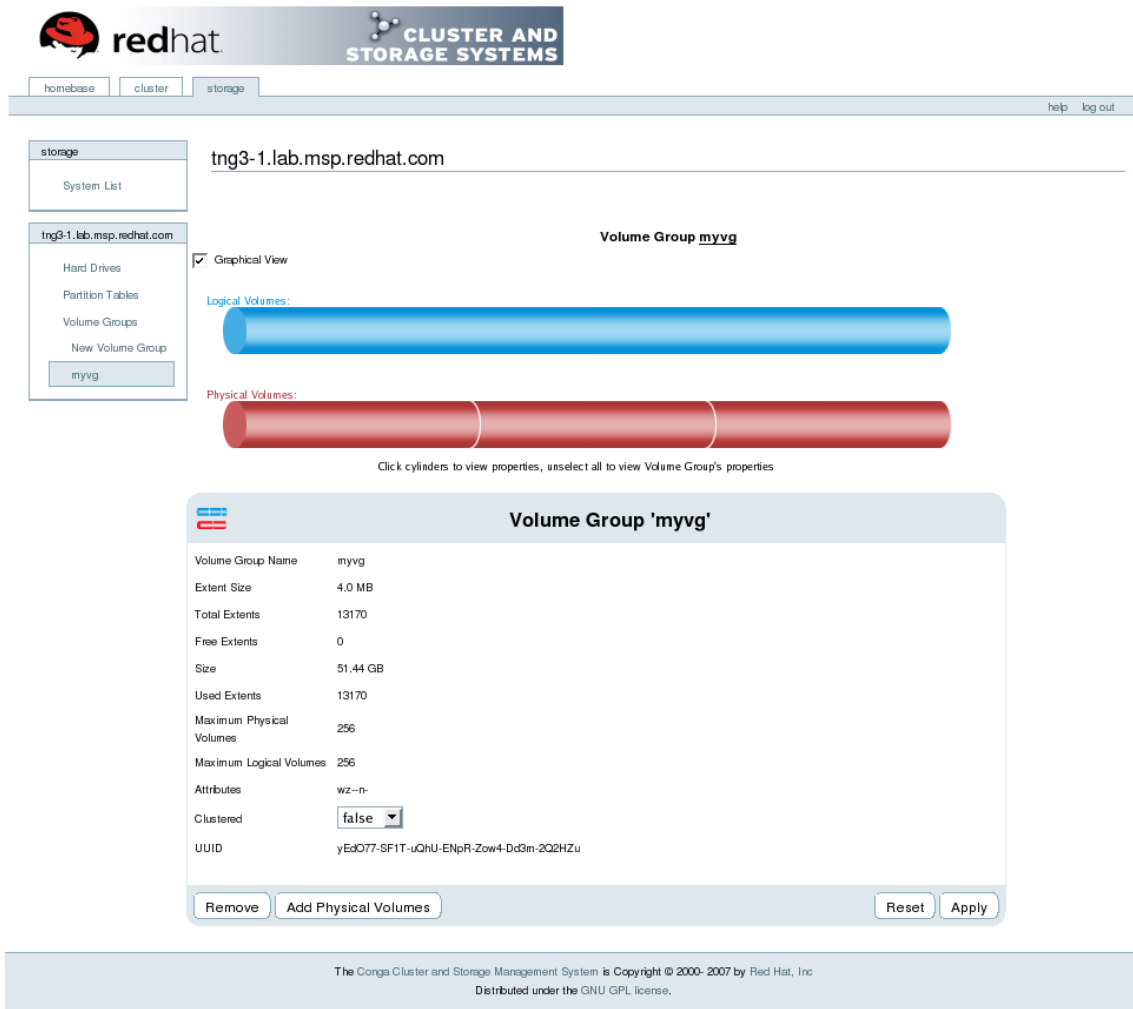


그림 1.16. Conga LVM Graphical User Interface

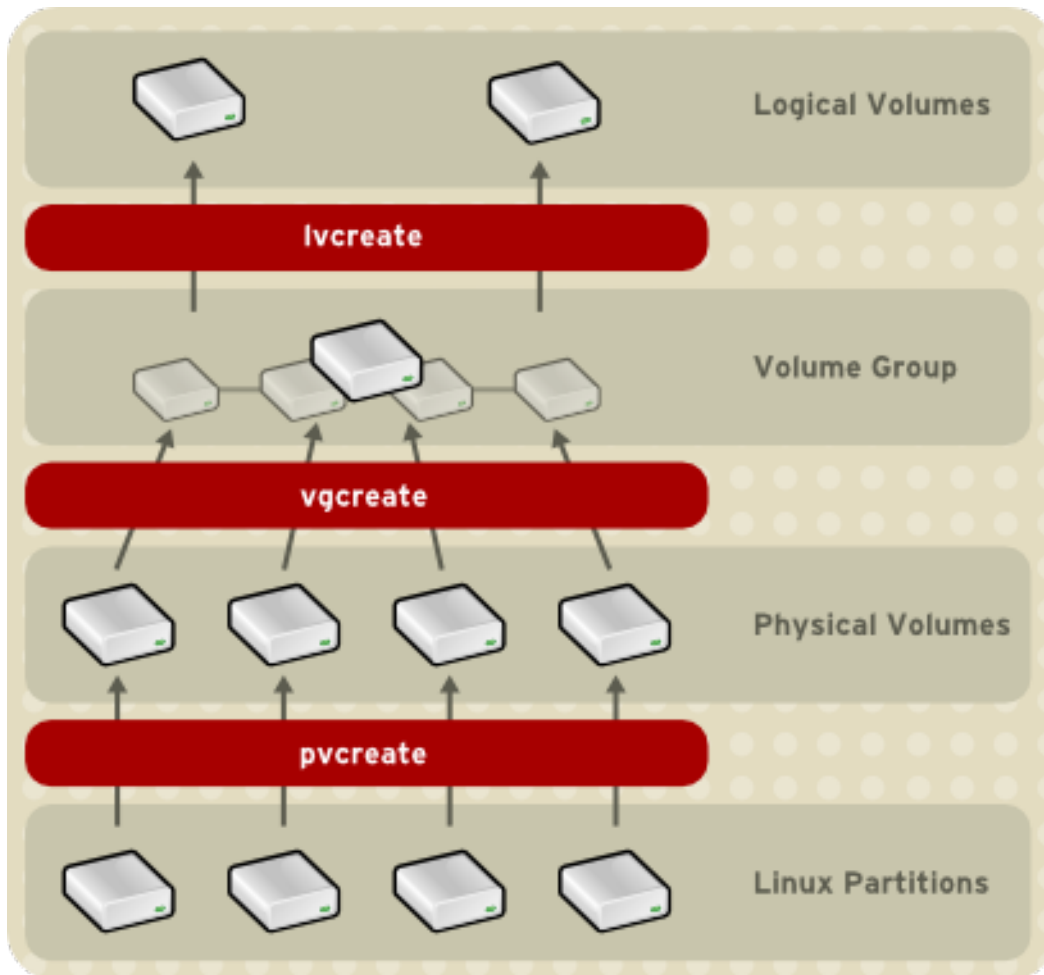


그림 1.17. Creating Logical Volumes

1.7. GNBD (GLOBAL NETWORK BLOCK DEVICE)

GNBD (Global Network Block Device)는 TCP/IP를 통해 Red Hat GFS로 블록 장치 액세스를 제공합니다. GNBD는 NBD와 개념이 비슷하지만 GNBD는 GFS 특정, GFS 만을 사용하여 독자적으로 실행됩니다. GNBD는 보다 강력한 기술이 – Fibre Channel이나 단일 개시자 SCSI – 필요하지 않거나 비용 제한이 있을 경우 유용합니다.

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 [그림 1.18. “GNBD 개요”](#). 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.

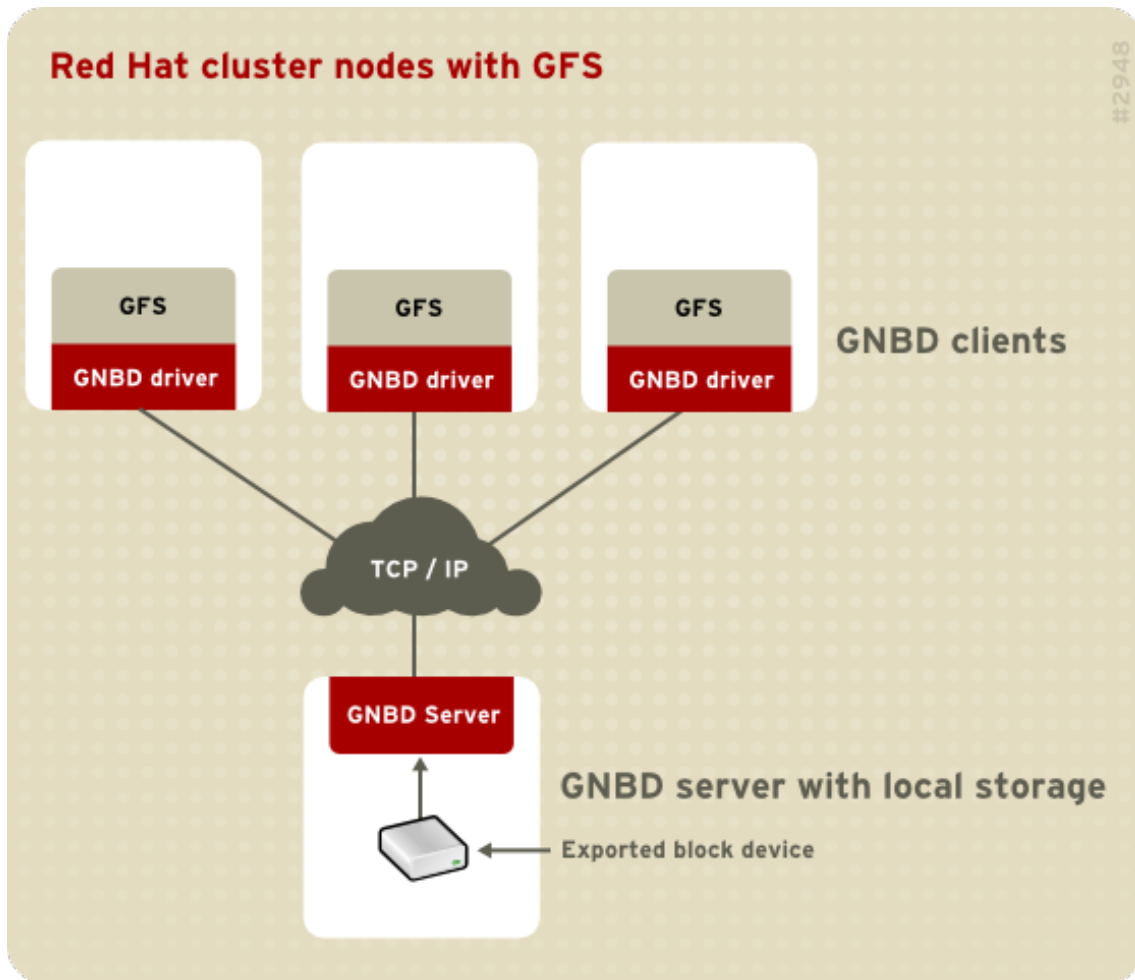


그림 1.18. GNBD 개요

1.8. LVS (LINUX VIRTUAL SERVER)

LVS (Linux Virtual Server)는 실제 서버를 통해 IP 로드 밸런스를 맞추기 위한 통합된 소프트웨어 구성 요소 모음입니다. LVS는 동일하게 설정된 한 쌍의 컴퓨터 상에서 실행됩니다: 즉 하나는 활성 LVS 라우터이고 다른 하나는 백업 LVS 라우터이어야 합니다. 활성 LVS 라우터는 두가지 역할을 수행합니다:

- 실제 서버를 통한 로드 밸런스
- 각각의 실제 서버에서의 서비스 무결성 확인

백업 LVS 라우터는 활성 LVS 라우터를 모니터링하고 활성 LVS 라우터에 문제가 발생할 경우 백업됩니다.

[그림 1.19. "Components of a Running LVS Cluster"](#) provides an overview of the LVS components and their interrelationship.

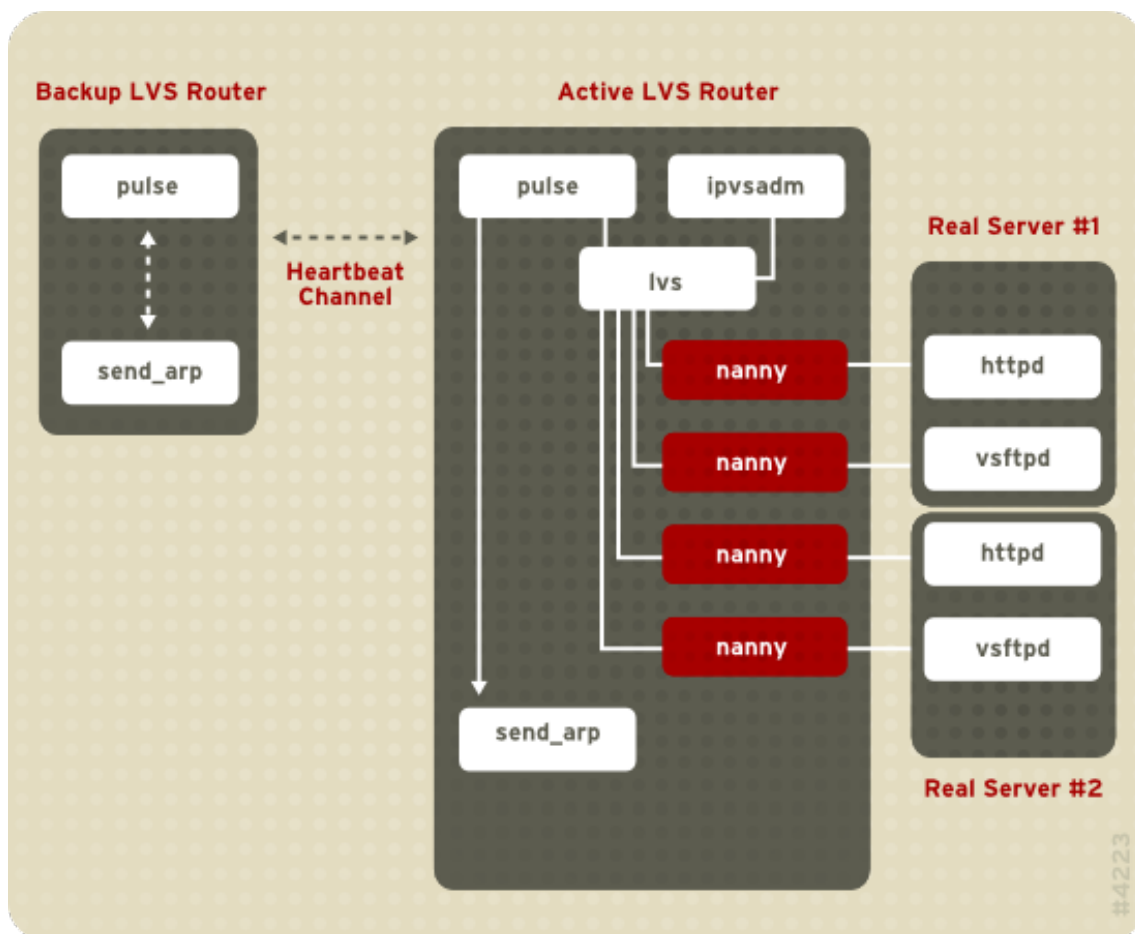


그림 1.19. Components of a Running LVS Cluster

`pulse` 데몬은 활성 및 비활성 LVS 라우터에서 실행됩니다. 백업 LVS 라우터에서 `pulse`는 활성 라우터의 공개 인터페이스로 `heartbeat`를 보내어 활성 LVS 라우터가 올바르게 작동하는 지를 확인합니다. 활성 LVS 라우터에서 `pulse`는 `lvs` 데몬을 시작하고 백업 LVS 라우터에서의 `heartbeat` 쿼리에 응답합니다.

일단 `lvs` 데몬이 시작되면 이는 `ipvsadm` 유틸리티를 호출하여 커널에 있는 IPVS (IP Virtual Server) 라우팅 테이블을 설정 및 관리하고 각각의 실제 서버에 있는 설정된 가상 서버에 필요한 `nanny` 프로세스를 시작합니다. 각각의 `nanny` 프로세스는 하나의 실제 서버에서 설정된 서비스의 상태를 확인하고 실제 서버에 있는 서비스가 잘 작동하지 않을 경우 `lvs` 데몬에 보고합니다. 작동 불량 사항이 발견되면, `lvs` 데몬은 `ipvsadm`에 지시하여 IPVS 라우팅 테이블에서 실제 서버를 삭제합니다.

백업 LVS 라우터가 활성 LVS 라우터에서 응답을 받지 못하면, 모든 가상 IP 주소를 백업 LVS 라우터의 NIC 하드웨어 주소 (MAC 주소)로 다시 지정하기 위해 `send_arp`를 호출하여 장애조치를 시작하고, 활성 LVS 라우터에 있는 `lvs` 데몬을 종료하기 위해 공개 및 개인 네트워크 인터페이스를 통해 활성 LVS 라우터에 명령을 보낸 후, 설정된 가상 서버의 요청을 수락하기 위해 백업 LVS 라우터에 있는 `lvs` 데몬을 시작합니다.

호스트 서비스 (예: 웹사이트 또는 데이터베이스 어플리케이션)에 액세스하는 외부 사용자에게 LVS는 하나의 서버로 나타납니다. 하지만, 사용자는 실제로 LVS 라우터 뒤에 있는 실제 서버에 액세스하게 됩니다.

실제 서버에 있는 데이터를 공유하기 위해 LVS에 내장된 구성 요소가 없기 때문에 다음과 같은 두가지 옵션이 필요합니다:

- 실제 서버를 통한 데이터 동기화
- 공유된 데이터 액세스에 필요한 토폴로지에 세번째 레이어 추가

첫 번째 옵션은 다수의 사용자가 실제 서버에서 업로드하거나 데이터를 변경하는 것을 허용하지 않는 서버에 적합합니다. 실제 서버에서 다수의 사용자가 데이터를 수정할 수 있게 할 경우 (예: 전자 상거래 웹사이트) 세번째 레이어를 추가하는 것이 좋습니다.

실제 서버에서 데이터를 동기화하는 방법에는 여러 가지가 있습니다. 예를 들어, 셸 스크립트를 사용하여 동시에 실제 서버에 업데이트된 웹 페이지를 게시할 수 있습니다. 또한, **rsync**와 같은 프로그램을 사용하여 일정 간격으로 모든 노드에 걸쳐 변경된 데이터를 복사할 수 있습니다. 하지만, 사용자가 수시로 파일을 업로드하거나 데이터베이스를 처리해야 하는 상황에서 데이터 동기화를 위해 스크립트를 사용하거나 또는 **rsync**을 사용하는 것은 동기화 기능을 최적화하지 못합니다. 따라서, 과중한 업로드, 데이터 베이스 처리량, 소용량이 있는 실제 서버에서는 데이터 동기화를 위해 **3-tier 토폴로지**를 사용하는 것이 좋습니다.

1.8.1. Two-Tier LVS Topology

그림 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 그림 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.

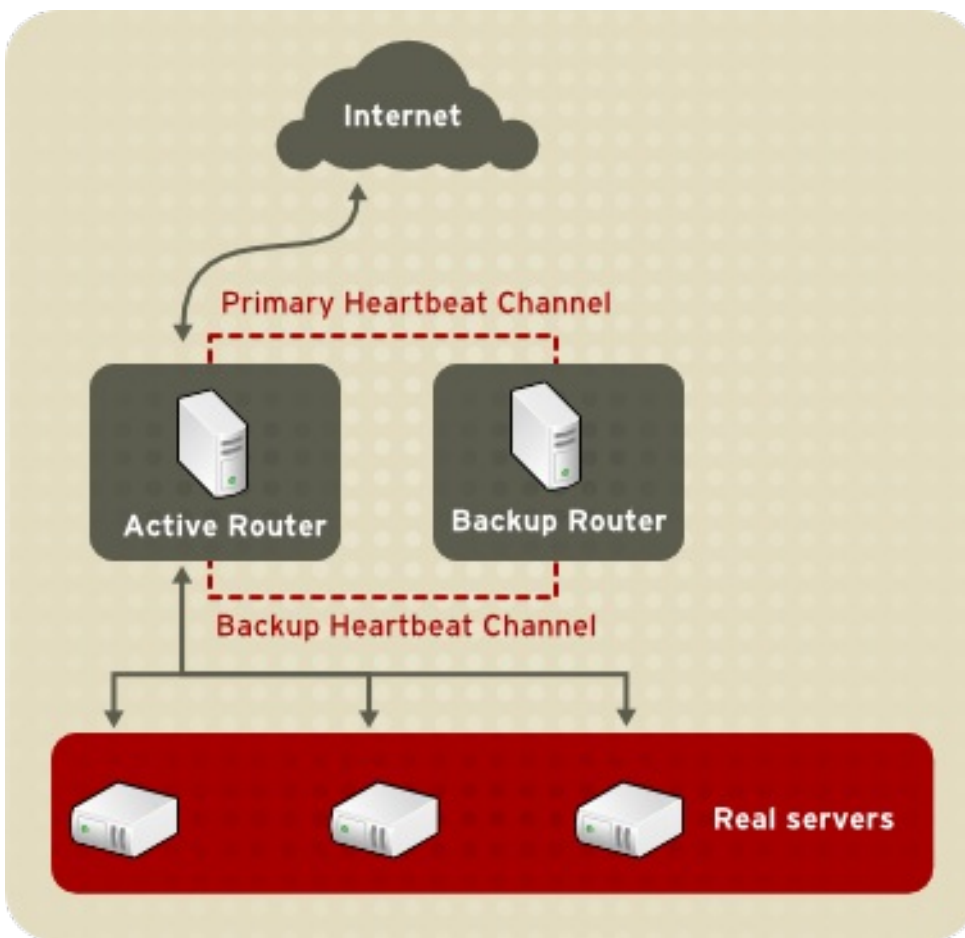


그림 1.20. Two-Tier LVS Topology

LVS 라우터로 들어온 서비스 요청은 가상 IP 주소 또는 VIP에서 인식됩니다. 이는 공개적으로 라우팅할 수 있는 주소로 사이트의 관리자가 `www.example.com`와 같은 FQDN (fully-qualified domain name)으로 연관지어 하나 이상의 가상 서버로 지정합니다. [1] VIP 주소는 장애 조치를 하는 동안 하나의 LVS 라우터에

서 다른 라우터로 이전시키게 되므로 해당 IP 주소로 현재 위치를 유지하게 됩니다. 이를 *유동 IP* 주소라고 합니다.

VIP 주소는 LVS 라우터를 공개 네트워크로 연결시키는 장치와 같은 것으로 알리아스될 수 있습니다. 예를 들어, `eth0`가 인터넷에 연결되어 있을 경우, 여러 가상 서버는 `eth0:1`로 알리아스될 수 있습니다. 다른 방법으로 각각의 가상 서버는 서비스마다 분리된 장치와 연관시킬 수 있습니다. 예를 들어, HTTP 소통량은 `eth0:1`에서 처리될 수 있으며, FTP 소통량은 `eth0:2`에서 처리될 수 있습니다.

한번에 하나의 LVS 라우터만이 활성화됩니다. 활성화된 LVS 라우터의 역할은 가상 IP 주소에서 실제 서버로 서비스 요청을 보내는 것입니다. 이는 여덟개의 로드 밸런싱 알고리즘 중 하나에 기반합니다:

- 라운드 로빈 (Round-Robin) 스케줄링 – 각각의 요청을 실제 서버의 풀로 순차적으로 배분하는 방식입니다. 이러한 알고리즘을 사용하여 용량이나 부하량을 고려하지 않고 모든 실제 서버를 동일하게 다룹니다.
- 가중치 기반 라운드 로빈 (Round-Robin) 스케줄링 – 각각의 요청을 실제 서버의 풀로 순차적으로 배분하는 방식이지만 거대한 처리 용량으로 서버에 더 많은 작업을 배분합니다. 처리 용량은 사용자 지정 가중치 요인에 의해 나타나며 동적 부하 정보에 의해 상하로 조정됩니다. 이는 서버 풀에 있는 실제 서버의 용량이 다를 경우 선택할 수 있는 사항입니다. 하지만, 요청 부하량이 아주 다양할 경우, 더 가중치가 있는 서버가 서버의 공유 또는 요청량보다 더 많이 응답할 수 있습니다.
- 최소 접속 (Least-Connection) 스케줄링 – 가장 접속이 적은 실제 서버로 더 많은 요청을 배분하는 방식입니다. 이는 동적인 스케줄링 알고리즘 유형 중 하나로, 요청 부하에 극도로 다양한 처리량이 있을 경우 유용합니다. 각각의 서버 노드에 대략 비슷한 처리 용량이 있는 실제 서버 풀의 경우에도 적합합니다. 실제 서버가 다양한 처리 용량이 있고, 가중치 기반 최소 접속 스케줄링일 경우 아주 적합합니다.
- 가중치 기반 최소 접속 (Weighted Least-Connections) 스케줄링 (기본값) – 처리 용량과 관련하여 가장 접속이 적은 서버로 더 많은 요청을 배분하는 방식입니다. 처리 용량은 사용자 지정 가중치 요인에 의해 나타나며, 동적 부하 정보에 의해 상하로 조정됩니다. 실제 서버 풀에 다양한 처리 용량의 하드웨어가 있을 경우 이러한 알고리즘이 적합합니다.
- LBLC (Locality-Based Least-Connection) 스케줄링 – 목적지 IP와 관련하여 가장 접속이 적은 서버로 더 많은 요청을 배분하는 방식입니다. 이러한 알고리즘은 프록시-캐쉬 서버 클러스터에서 사용됩니다. 이는 서버가 서버 처리 용량을 초과하지 않고 반부하에 서버가 있지 않을 경우 IP 주소를 최소로 부하되는 실제 서버로 할당하게 되어 IP 주소에 대한 패킷을 해당 주소에 대한 서버로 라우팅합니다.
- LBLCR (Locality-Based Least-Connection with Replication) 스케줄링 – 목적지 IP와 관련하여 가장 접속이 적은 서버로 더 많은 요청을 배분하는 방식입니다. 이러한 알고리즘은 프록시-캐쉬 서버 클러스터에 사용됩니다. 대상 IP 주소를 실제 서버 노드의 서브셋으로 묶는다는 점에서 LBLC 스케줄링과 다릅니다. 이는 서브셋에 있는 가장 접속이 적은 서버로 요청을 라우트합니다. 목적지 IP 주소에 대해 모든 노드가 처리 용량을 초과할 경우, 가장 접속이 적은 실제 서버를 실제 서버의 전체 풀에서 목적지 IP의 실제 서버에 대한 서브셋으로 추가하여 해당 목적지 IP 주소에 대한 새로운 서버를 복사합니다. 최고로 과부화된 노드는 실제 서버 서브셋에서 빠지게 됩니다.
- 소스 해시 (Source Hash) 스케줄링 – 정적 해시 테이블에 있는 소스 IP를 찾아 실제 서버의 풀로 요청을 배분하는 방식입니다. 이러한 알고리즘은 여러 방화벽이 있는 LVS 라우터에 사용됩니다.

또한, 활성화된 LVS 라우터는 *send/expect 스크립트*를 통해 실제 서버에 있는 특정 서비스의 전반적인 상태를 정적으로 모니터링합니다. HTTPS나 SSL과 같은 정적인 데이터에 필요한 서비스의 상태를 점검하기 위해 외부에서 실행 가능하게 할 수 있습니다. 실제 서버에서의 서비스가 잘 작동하지 않을 경우, 활성화된 LVS 라우터는 서버가 정상적인 작동 상태로 돌아올 때까지 실제 서버로의 작업 전송을 중단합니다.

백업 LVS 라우터는 대기 시스템의 역할을 실행합니다. 주기적으로 LVS 라우터는 주요 외부 공개 인터페이스, 장애 조치 상태, 개인 인터페이스를 통해 하트비트 메시지 (heartbeat message)를 교환합니다. 백업 LVS 라우터는 정해진 간격 내로 하트비트 메시지를 받지 못하면, 장애 조치를 초기화하고 활성화된 LVS

라우터의 역할을 실행합니다. 장애 조치가 진행되는 동안 백업 LVS 라우터는 *ARP spoofing*이라는 기술을 사용하여 장애가 발생한 라우터에 의해 VIP 주소를 전달 받게 됩니다. – 여기서 백업 LVS 라우터는 IP 패킷의 목적지를 장애가 발생한 노드로 지정했음을 알립니다. 장애가 발생한 노드가 정상으로 돌아오면, 백업 LVS 라우터는 다시 백업 역할을 실행하게 됩니다.

The simple, two-tier configuration in [그림 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

[그림 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.

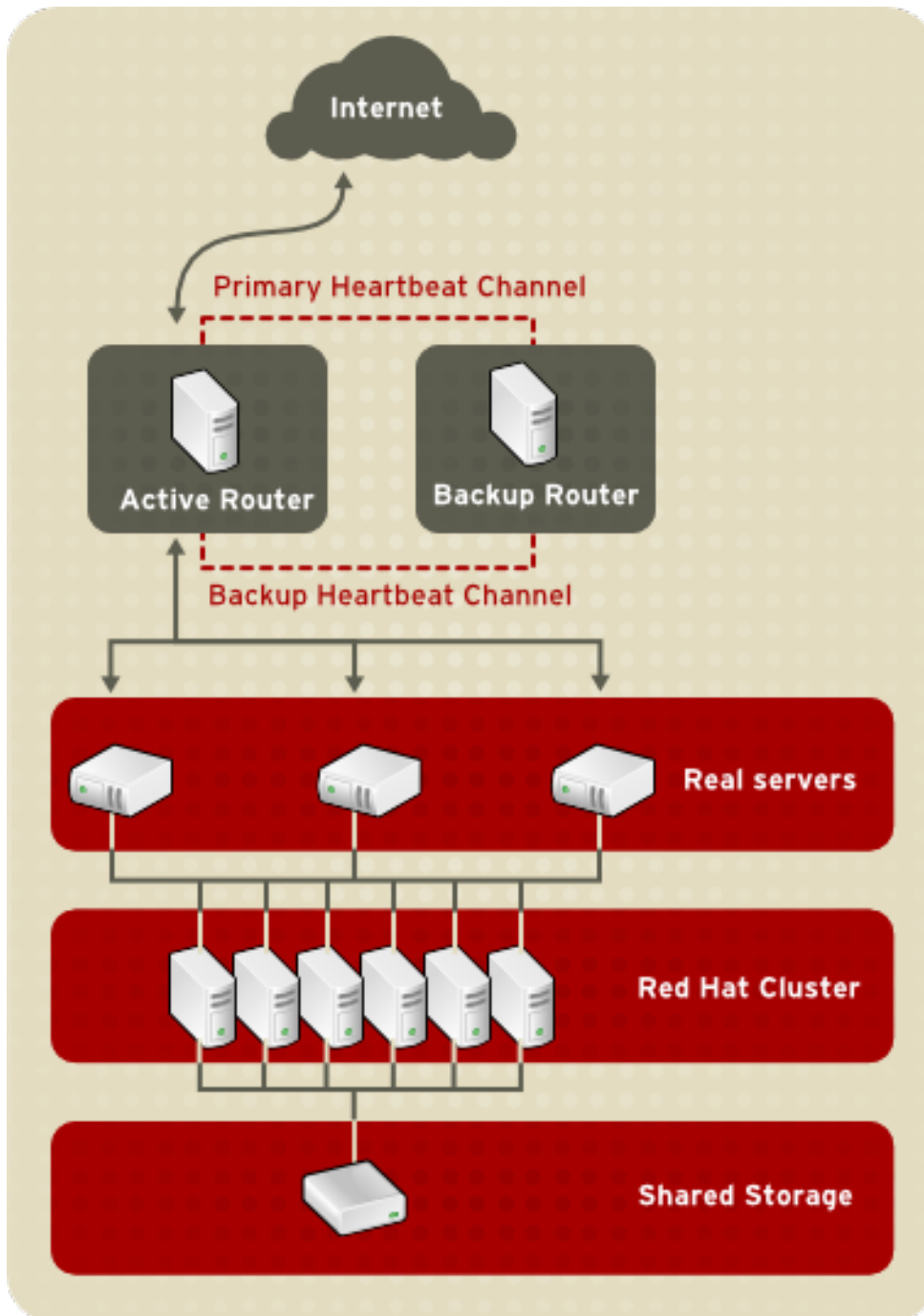


그림 1.21. Three-Tier LVS Topology

토폴로지는 소용량이 많은 FTP 서버에 아주 적합합니다, 여기서 액세스할 수 있는 데이터는 중앙고가용성 서버에 저장되어 있으며 내보내어진 NFS 디렉토리나 Samba 공유 디렉토리를 통해 각각의 실제 서버에 액세스할 수 있습니다. 이러한 토폴로지는 트랜잭션을 위한 중앙고가용성 데이터베이스를 액세스하는 웹사이트에 사용할 것을 권장합니다. 또한, Red Hat 클러스터를 사용한 active-active 설정으로, 하나의고가용성 클러스터를 설정하여 동시에 두가지 역할을 실행하게 할 수 있습니다.

1.8.3. 라우팅 방식

NAT (Network Address Translation) 라우팅 또는 LVS를 사용한 직접 라우팅을 사용할 수 있습니다. 다음 부분에서는 NAT 라우팅과 LVS를 사용한 직접 라우팅에 대해 간략하게 설명합니다.

1.8.3.1. NAT 라우팅

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

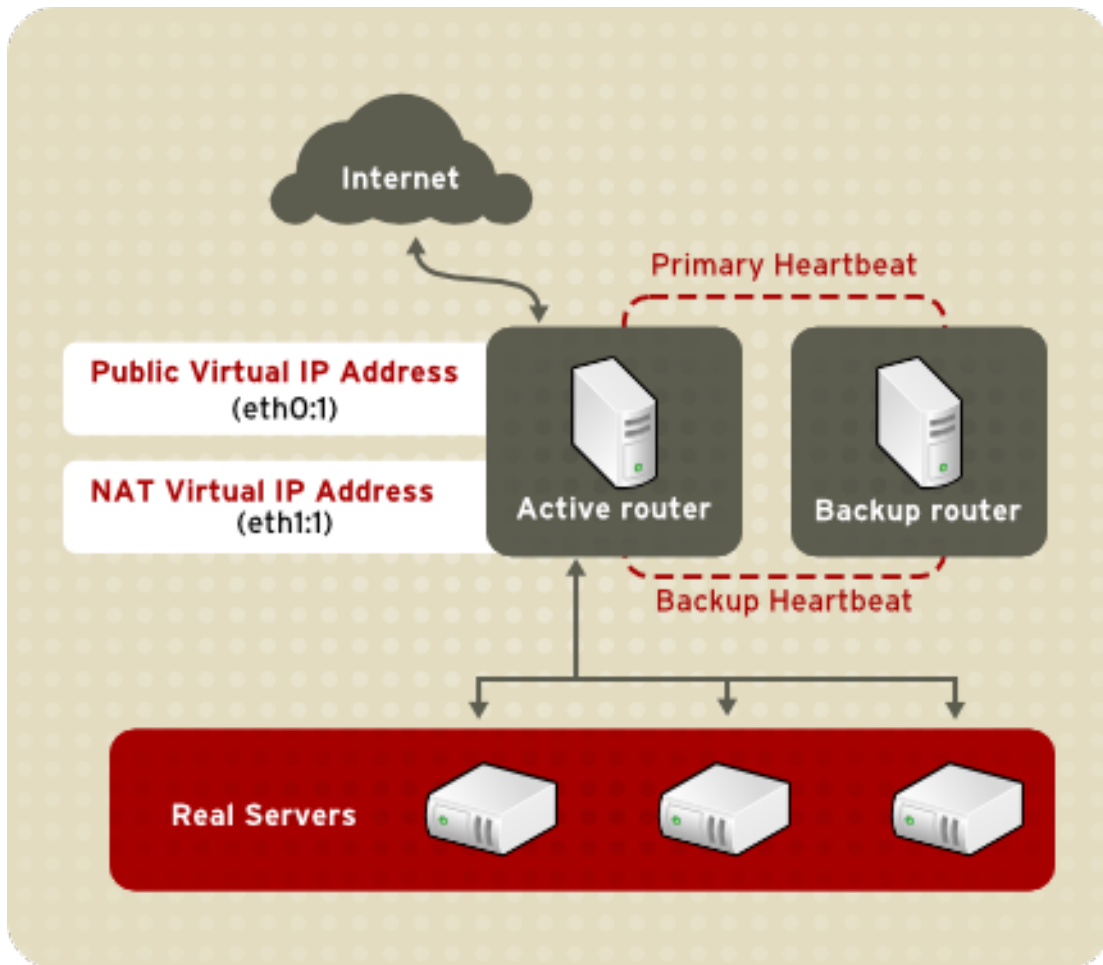


그림 1.22. LVS Implemented with NAT Routing

예에서 활성화된 LVS 라우터에는 두개의 NIC가 있습니다. 인터넷용 NIC는 eth0에 실제 IP 주소가 있고 알리아스된 eth0:1에는 유동 IP 주소가 있습니다. 개인 네트워크 인터페이스 용 NIC에는 eth1에 실제 IP 주소가 있고 알리아스된 eth1:1에는 유동 IP 주소가 있습니다. 장애 조치 시, 인터넷에 대한 가상 인터페이스와 가상 인터페이스에 대한 개인 네트워크는 동시에 백업 LVS 라우터에 의해 백업됩니다. 개인 네트워크에 있는 모든 실제 서버는 NAT 라우터에 대해 활성화된 LVS 라우터와 통신하기 위한 기본값 라우터로 유동 IP 주소를 사용하므로 인터넷에서의 요청에 응답하는 기능에 문제를 일으키지 않습니다.

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.

이러한 토폴로지를 사용하여 활성화된 LVS 라우터에서는 요청을 받아 적절한 서버로 보냅니다. 그 뒤 실제 서버는 요청을 처리하고 LVS 라우터로 패킷을 되돌려 보냅니다. LVS 라우터는 네트워크 주소 트랜잭션을 사용하여 패킷에 있는 실제 서버의 주소를 LVS 라우터 공개 VIP 주소로 교체합니다. 실제 서버의 실질적인 IP 주소가 요청 중인 클라이언트에게 보이지 않게 되므로 이러한 절차를 IP 마스커레이딩이라고 부릅니다.

NAT 라우팅을 사용하여, 다양한 운영 체제가 실행되고 있는 아무 컴퓨터나 실제 서버가 될 수 있습니다. NAT 라우팅의 주요 취약점은 들어오고 나가는 모든 요청을 처리해야 하기 때문에 LVS 라우터의 대량의 분배 작업에서 병목 현상이 일어날 수 있다는 점입니다.

1.8.3.2. 직접 라우팅

직접 라우팅은 NAT 라우팅과 비교하여 성능이 향상되었습니다. 직접 라우팅으로 실제 서버는 LVS 라우팅을 통해 외부로 나가는 패킷을 전달하지 않고 요청 중인 사용자에게 직접 패킷을 처리하고 보냅니다. 직접 라우팅은 LVS 라우터의 작업을 들어오는 패킷이 처리하게 하여 네트워크 성능 문제가 발생하지 않게 합니다.

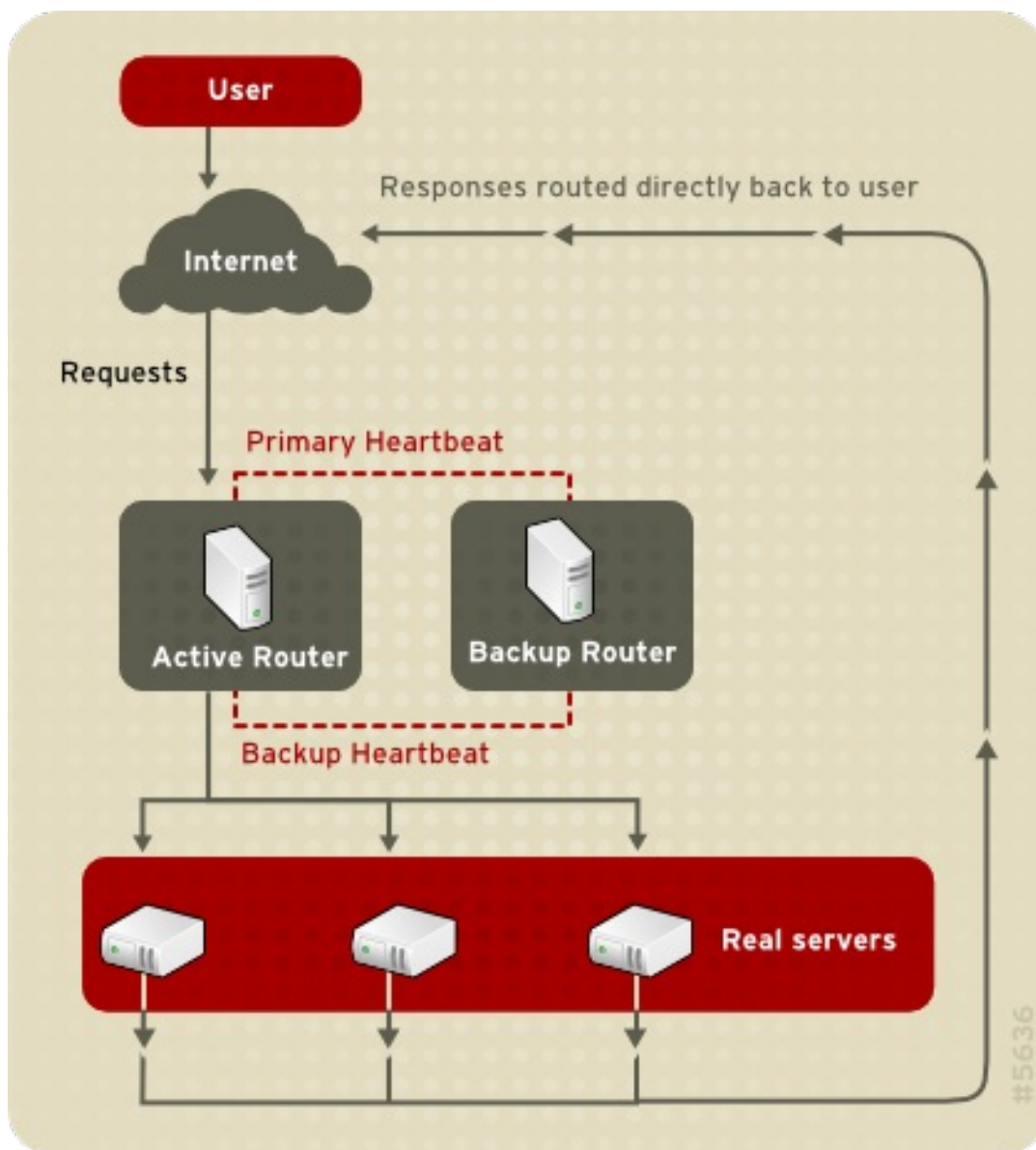


그림 1.23. LVS Implemented with Direct Routing

전형적인 직접 라우팅 LVS 설정에서, LVS 라우터는 가상 IP (VIP)를 통해 들어오는 서버 요청을 받고 스케줄링 알고리즘을 사용하여 실제 서버에 요청을 보냅니다. 각각의 실제 서버는 요청을 처리하고 LVS 라우터를 바이패싱하여 클라이언트에게 직접 응답합니다. LVS 라우터에 추가된 처리량 없이 실제 서버에서 클라이언트로 나가는 패킷을 보내기 위해 실제 서버에서 확장성을 감안한 직접 라우팅이 추가될 수 있어 과중한 네트워크 로드에서 병목 현상이 나타날 수 있습니다.

LVS 에서 직접 라우팅을 사용하는데 있어서 여러 장점이 있는 반면 제한점도 있습니다. 직접 라우팅과 LVS에서 가장 일반적으로 발생하는 문제는 ARP (Address Resolution Protocol)과의 문제입니다.

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.

직접 라우팅 LVS 설정에 있는 ARP 요청에서의 문제는 IP 주소로의 클라이언트 요청이 처리될 요청에 대한 MAC 주소와 관련되어 있어야 하며, LVS 라우터의 가상 IP 주소도 MAC 주소와 관련되어 있어야 한다는 점입니다. 하지만, LVS 라우터와 실제 서버의 VIP가 같을 경우, ARP 요청은 VIP와 관련되어 있는 모든 노드로 브로드캐스트됩니다. 이는 VIP가 실제 서버 중 하나에 관련되어 직접적으로 요청을 처리하는 것, LVS 라우터를 바이패싱하여 LVS 설정 목적을 의미없게 하는 것과 같이 여러가지 문제를 발생시킬 수 있습니다. 클라이언트 요청에 빠르게 응답할 수 있는 강력한 CPU와 함께 LVS 라우터를 사용하는 것으로 이러한 문제를 해결할 수 없습니다. LVS 라우터가 과부하 상태일 경우, 충분히 활용되지 않은 실제 서버보다 더 늦게 ARP 요청에 응답하게 될 것입니다. 충분히 활용되지 않은 실제 서버는 보다 빨리 응답하고 요청 중인 클라이언트의 ARP 캐시에 VIP를 지정하게 됩니다.

이러한 문제를 해결하기 위해 들어오는 요청은 오직 VIP만을 LVS 라우터에 연관시켜서 요청을 처리하고 이를 실제 서버 풀에 보내도록 합니다. 이러한 작업은 **arptables** 패킷 필터링 도구를 사용하여 실행합니다.

1.8.4. 지속성 및 방화벽 표시 기능

특정한 경우에는 로드 밸런스 알고리즘이 가장 적합한 서버에 요청을 보내는 것 보다 동일한 실제 서버에 반복적으로 다시 연결하는 것이 클라이언트에게 더 좋을 수 있습니다. 예를 들어, 여러 화면의 웹 형식, 쿠키, SSL, FTP 연결일 경우에 그러합니다. 이러한 경우, 문맥을 유지하기 위해 트랜잭션이 같은 서버에 의해 처리되지 않는 한 클라이언트가 제대로 작업할 수 없게 됩니다. LVS는 **지속성 (persistence)** 및 **방화벽 표시 기능 (firewall marks)**이라는 두가지 다른 종류의 기능을 사용하여 이를 처리합니다.

1.8.4.1. Persistence

지속성이 활성화되면 이는 타이머와 같은 기능을 수행합니다. 클라이언트가 서비스에 연결할 경우, LVS는 특정 기간에 대한 마지막 연결을 기억하게 됩니다. 같은 클라이언트 IP가 특정 기간 내에 다시 연결될 경우 이전에 연결했던 것과 같은 서버로 전달됩니다. - 로드 밸런스 메카니즘 바이패싱. 시간 창 외부에서 연결할 경우 스케줄링 규칙에 따라 처리됩니다.

지속성 기능은 어떤 주소가 더 높은 수준의 지속성을 갖게 할 지를 제어하기 위한 도구로서 클라이언트 IP 주소 테스트를 위한 서브넷 마스크를 지정하게 하여 서브넷에 그룹으로 연결합니다.

목적지가 다른 포트에 대한 그룹 연결은 FTP와 같이 하나 이상의 포트를 사용하여 통신하는 프로토콜의 경우 중요한 사항이 될 수 있습니다. 하지만, 목적지가 다른 포트에 되어있는 그룹 연결 문제를 처리하는데 지속성 기능이 가장 효과적인 방법은 아닙니다. 이러한 경우, **방화벽 표시 기능 (firewall marks)**을 사용하는 것이 가장 좋습니다.

1.8.4.2. 방화벽 표시 기능

방화벽 표시 기능은 연관된 프로토콜의 프로토콜이나 그룹에 사용되는 가장 쉽고 효과적인 그룹 포트 방법입니다. 예를 들어, LVS가 전자 상거래 사이트를 실행하도록 되어 있을 경우, 방화벽 표시 기능은 포트 80에서의 HTTP 연결과 포트 443에서의 HTTPS 연결을 일괄적으로 묶어 처리하는데 사용될 수 있습니다. 각각의 프로토콜에 대해 가상 서버에 동일한 방화벽 표시 기능을 지정하면 포트가 연결된 후 LVS 라우터가 동일한 실제 서버에 모든 요청을 보내기 때문에 트랜잭션에 대한 상태 정보를 보호할 수 있습니다.

이와 같은 효과적이고 사용 간편함 때문에 LVS 관리자는 그룹 연결을 할 때 마다 지속성 기능보다 방화벽 표시 기능을 사용해야 합니다. 하지만, 클라이언트가 일정 기간 동안 동일한 가상 서버로 다시 연결하게 하기 위해 가상 서버에 방화벽 기능과 함께 지속성 기능을 추가시킬 수 있습니다.

1.9. 클러스터 관리 도구

Red Hat Cluster Suite은 Red Hat 클러스터 설정 및 관리를 위한 다양한 도구를 제공합니다. 다음 부분에서는 Red Hat Cluster Suite과 함께 사용할 수 있는 관리 도구에 대한 개요를 다루고 있습니다.

- 1.9.1절. “Conga”

- 1.9.2절. “클러스터 관리 GUI”
- 1.9.3절. “명령행 관리 도구”

1.9.1. Conga

Conga는 소프트웨어 구성 요소의 통합된 모음으로 Red Hat 클러스터 및 스토리지를 중앙 집중적으로 설정 및 관리합니다. **Conga**에는 다음과 같은 주요 기능이 있습니다:

- 클러스터 및 스토리지 관리에 필요한 하나의 웹 인터페이스
- 클러스터 데이터의 자동 디플로이 및 패키지 지원
- 기존 클러스터와 쉽게 통합
- 재인증할 필요가 없음
- 클러스터 상태 및 로그의 통합
- 사용자 권한을 통한 정교한 제어 기능

Conga의 주요 구성 요소에는 **luci** 및 **ricci**로 이는 각각 설치될 수 있습니다. **luci**는 하나의 컴퓨터 상에서 실행되는 서버로 **ricci**를 통해 다중 클러스터 및 컴퓨터와 통신합니다. **ricci**는 각각의 컴퓨터 (클러스터 멤버 또는 독립형 컴퓨터)에서 실행되는 에이전트로 **Conga**에 의해 관리됩니다.

luci는 웹 브라우저로 액세스할 수 있으며 세가지 주요 기능을 제공하며 이는 다음과 같은 탭을 통해 액세스할 수 있습니다:

- **homebase** – 컴퓨터 추가 또는 삭제, 사용자 추가 또는 삭제, 사용자 권한 설정을 위한 도구를 제공합니다. 시스템 관리자만이 이 탭에 액세스할 수 있습니다.
- **cluster** – 클러스터 생성 및 관리를 위한 도구를 제공합니다. **luci**의 각각의 인스턴스는 **luci**를 사용하여 설정된 클러스터를 나열합니다. 시스템 관리자는 이 탭에 나열된 모든 클러스터를 관리할 수 있습니다. 기타 다른 사용자는 관리 권한이 있는 클러스터만 관리할 수 있습니다. (관리자에 의해 권한을 받음).
- **storage** – 스토리지의 원격 관리를 위한 도구를 제공합니다. 이 탭에 있는 도구를 사용하여 스토리지의 클러스터 소속 여부에 상관 없이 컴퓨터 상에 있는 스토리지를 관리할 수 있습니다.

클러스터 또는 스토리지를 관리하기 위해, 관리자는 **luci** 서버에 클러스터 또는 컴퓨터를 추가 (또는 등록) 해야 합니다. 클러스터 또는 컴퓨터가 **luci**에 등록되면 각각의 컴퓨터에 대한 FQDN 호스트명 또는 IP 주소가 **luci** 데이터베이스에 저장됩니다.

다른 **luci** 인스턴스에서 하나의 **luci** 인스턴스에 대한 데이터베이스를 지정할 수 있습니다. 이러한 기능은 **luci** 서버 인스턴스를 복사하는 방법을 제공하며 효율적인 업그레이드 및 테스트 경로를 제공합니다. **luci** 인스턴스 설치 시 데이터 베이스는 비어있게 됩니다. 하지만 새로운 **luci** 서버를 지정할 때 기존의 **luci** 서버에서 모든 또는 일부분의 **luci** 데이터베이스를 가져올 수 있습니다.

각각의 **luci** 인스턴스는 초기 설치 – 관리에서 하나의 사용자를 갖습니다. 관리자는 **luci** 서버에 시스템을 추가할 수 있습니다. 또한 관리자는 추가 사용자 계정을 생성하고 사용자에게 대해 **luci** 데이터 베이스에 등록된 클러스터 및 컴퓨터 액세스 권한을 결정할 수 있습니다. 새로운 **luci** 서버에서 클러스터와 컴퓨터를 가져올 수 있는 것처럼 배치(batch) 옵션으로 사용자를 가져올 수 있습니다.

luci 서버에 관리할 컴퓨터를 추가할 때 인증 과정은 한 번 이루어 집니다. 그 이후로는 (인증서가 CA에 의해 취소되지 않는 한) 인증 과정이 필요하지 않게 됩니다. 다음으로 **luci** 사용자 인터페이스를 통해 클러스터 및 스토리지를 원격으로 설정 및 관리할 수 있습니다. **luci** 및 **ricci**는 XML을 통해 서로 통신합니다.

다음에서는 세 가지의 주요한 **luci** 탭을 보여주고 있습니다: **homebase**, **cluster**, **storage**.

Conga에 관한 보다 자세한 내용은 *Red Hat 클러스터 설정 및 관리*를 참조하시거나 **luci** 서버에 있는 온라인 도움말에서 참조하시기 바랍니다.

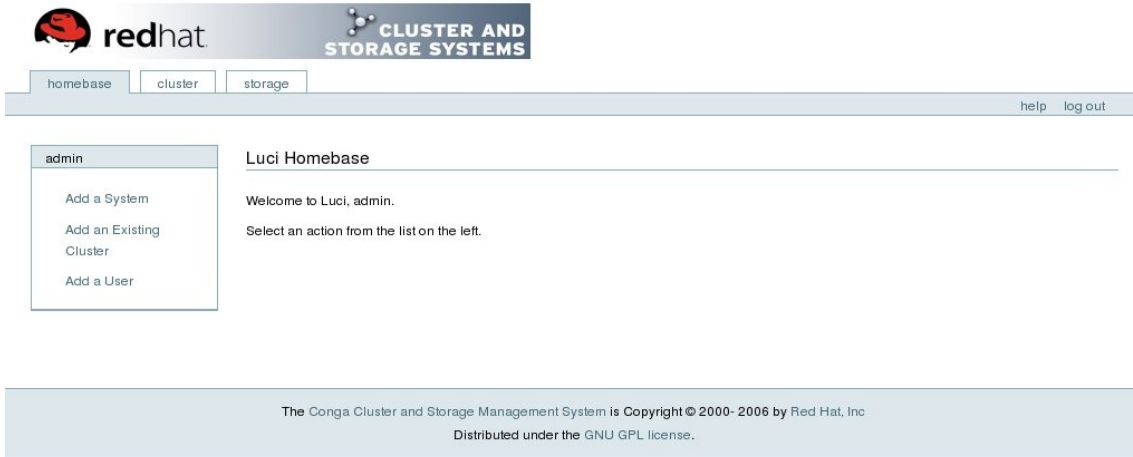


그림 1.24. luci homebase 탭

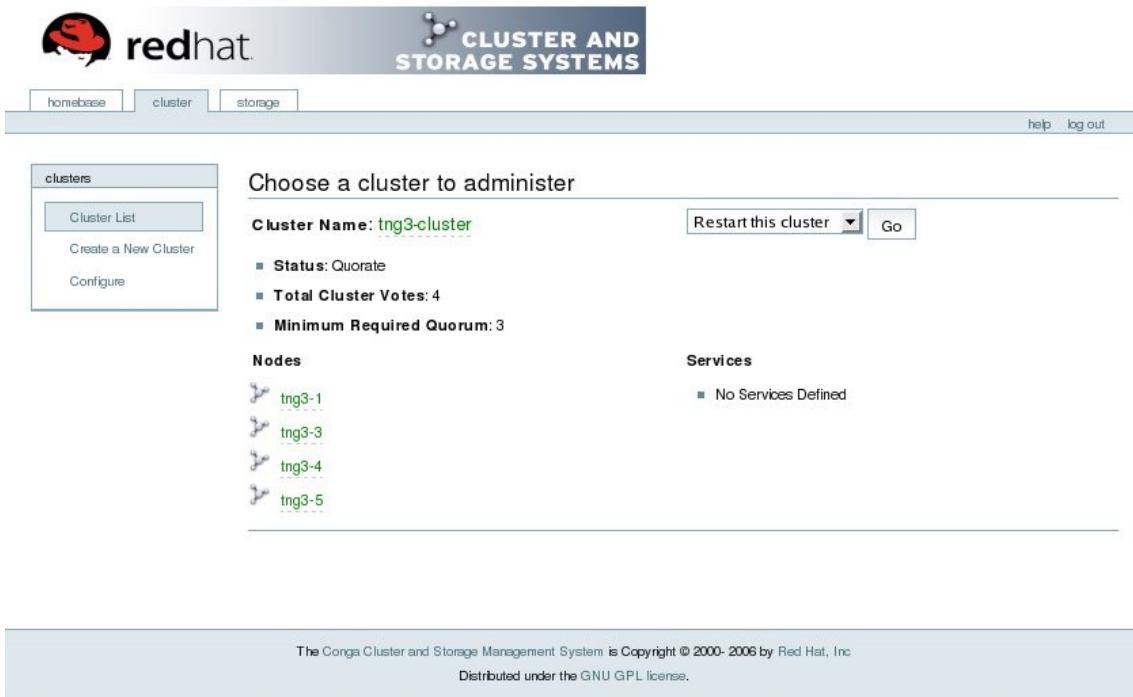


그림 1.25. luci cluster 탭

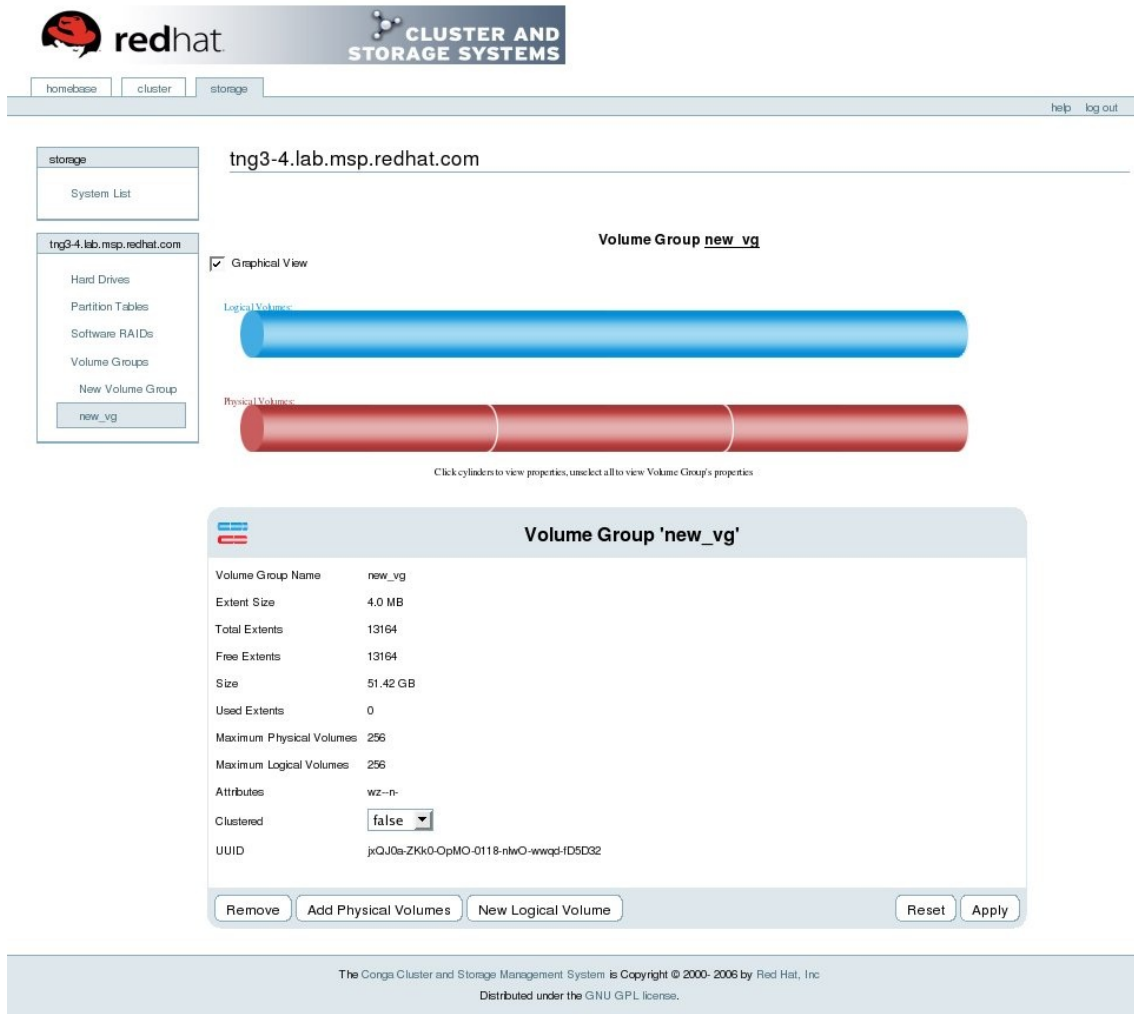


그림 1.26. luci storage 탭

1.9.2. 클러스터 관리 GUI

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 1.3절. “Cluster Infrastructure” and 1.4절. “고가용성 서비스 관리”). 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.

- 1.9.2.1절. “Cluster Configuration Tool”
- 1.9.2.2절. “Cluster Status Tool”

1.9.2.1. Cluster Configuration Tool

You can access the **Cluster Configuration Tool** (그림 1.27. “Cluster Configuration Tool”) through the **Cluster Configuration** tab in the Cluster Administration GUI.

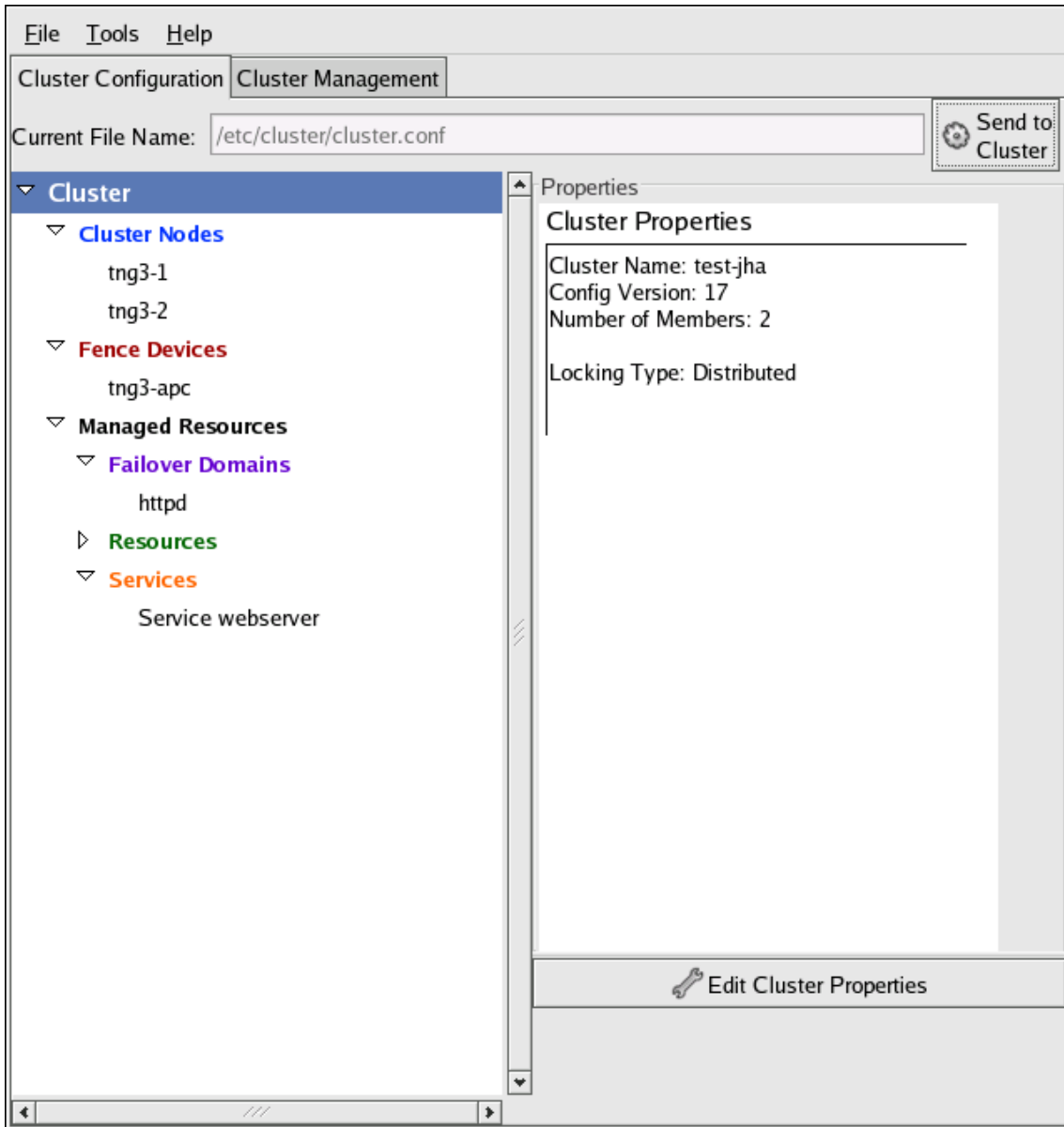


그림 1.27. Cluster Configuration Tool

Cluster Configuration Tool에서는 왼쪽 패널의 계층 구조로 된 그래픽 화면과 함께 설정 파일 (`/etc/cluster/cluster.conf`)에 있는 클러스터 설정 구성 요소를 보여줍니다. 구성 요소 이름의 왼쪽에 있는 삼각 아이콘은 하나 이상의 부수적 요소가 있는 구성 요소를 나타냅니다. 삼각 아이콘을 클릭하여 각 구성요소 아래의 세가지 부분을 펼쳐 볼 수 있습니다. 다음에서는 GUI에 있는 구성요소에 관한 요약 을 다루고 있습니다:

- **Cluster Nodes** – 클러스터 노드를 보여 줍니다. 노드는 **Cluster Nodes**에서 부속 요소로 노드 이름으로 나타납니다. 오른쪽 하단 프레임 (**Properties**아래)에 있는 설정 버튼을 사용하여 노드를 추가, 삭제할 수 있으며 노드 등록 정보를 편집하고 각 노드의 **fencing** 방법을 설정할 수 있습니다.
- **Fence Devices** – fence 장치를 보여줍니다. fence 장치는 **Fence Devices**에서 부속 요소로 나타납니다. 오른쪽 하단 프레임 (**Properties** 아래)에 있는 설정 버튼을 사용하여 fence 장치를 추가, 삭제할 수 있으며 fence 장치 등록정보를 편집할 수 있습니다. Fence 장치는 각각의 노드에 필요한fencing을 (**Manage Fencing For This Node** 버튼을 사용) 설정하기 전에 지정해야 합니다.
- **Managed Resources** – 장애 조치 도메인, 리소스, 서비스를 보여줍니다.
 - **Failover Domains** – 노드에 장애 발생 시 고가용성 서비스 실행에 사용할 한 개 이상의 클

러스토 노드 서브셋을 설정합니다. 장애 조치 도메인은 **Failover Domains** 아래에서 부수적인 요소로서 나타납니다. 오른쪽 하단 프레임 (**Properties** 아래)에 있는 설정 버튼을 사용하여 (**Failover Domains**가 선택되었을 경우) 장애 조치 도메인을 생성하거나 또는 (**failover domain**이 선택되었을 경우) 장애 조치 도메인 등록 정보를 편집할 수 있습니다.

- **Resources** – 고가용성 서비스로 사용할 공유 리소스를 설정합니다. 공유 리소스에는 파일 시스템, IP 주소, NFS 마운트 및 내보내기, 클러스터에 있는 모든 고가용성 서비스에서 사용 가능한 사용자 생성 스크립트가 들어 있습니다. 리소스는 **Resources** 하에서 부수적인 요소로 나타납니다. 오른쪽 하단 프레임 (**Properties** 아래)에 있는 설정 버튼을 사용하여 **Resources**가 선택되었을 경우) 리소스를 생성하거나 또는 (**resource**가 선택되었을 경우) 리소스 등록 정보를 편집할 수 있습니다.



참고

Cluster Configuration Tool 로 개인 리소스를 설정할 수 있습니다. 개인 리소스는 하나의 서버에서만 사용하도록 설정된 리소스입니다. GUI에 있는 **Service** 구성 요소에서 개인 리소스를 설정할 수 있습니다.

- **Services** – 고가용성 서비스를 생성하고 설정합니다. (공유 또는 개인) 리소스와 장애 조치 도메인을 할당하고 서비스에 필요한 복구 정책을 정의하는 것으로 서비스는 설정됩니다. 서비스는 **Services** 하의 부수적인 요소로 나타납니다. 오른쪽 하단 프레임 (**Properties** 아래)에 있는 설정 버튼을 사용하여 (**Services**가 선택되었을 경우) 서비스를 생성하거나 또는 (**service**가 선택되었을 경우) 서비스 등록 정보를 편집할 수 있습니다.

1.9.2.2. Cluster Status Tool

You can access the **Cluster Status Tool** ([그림 1.28. “Cluster Status Tool”](#)) through the **Cluster Management** tab in Cluster Administration GUI.

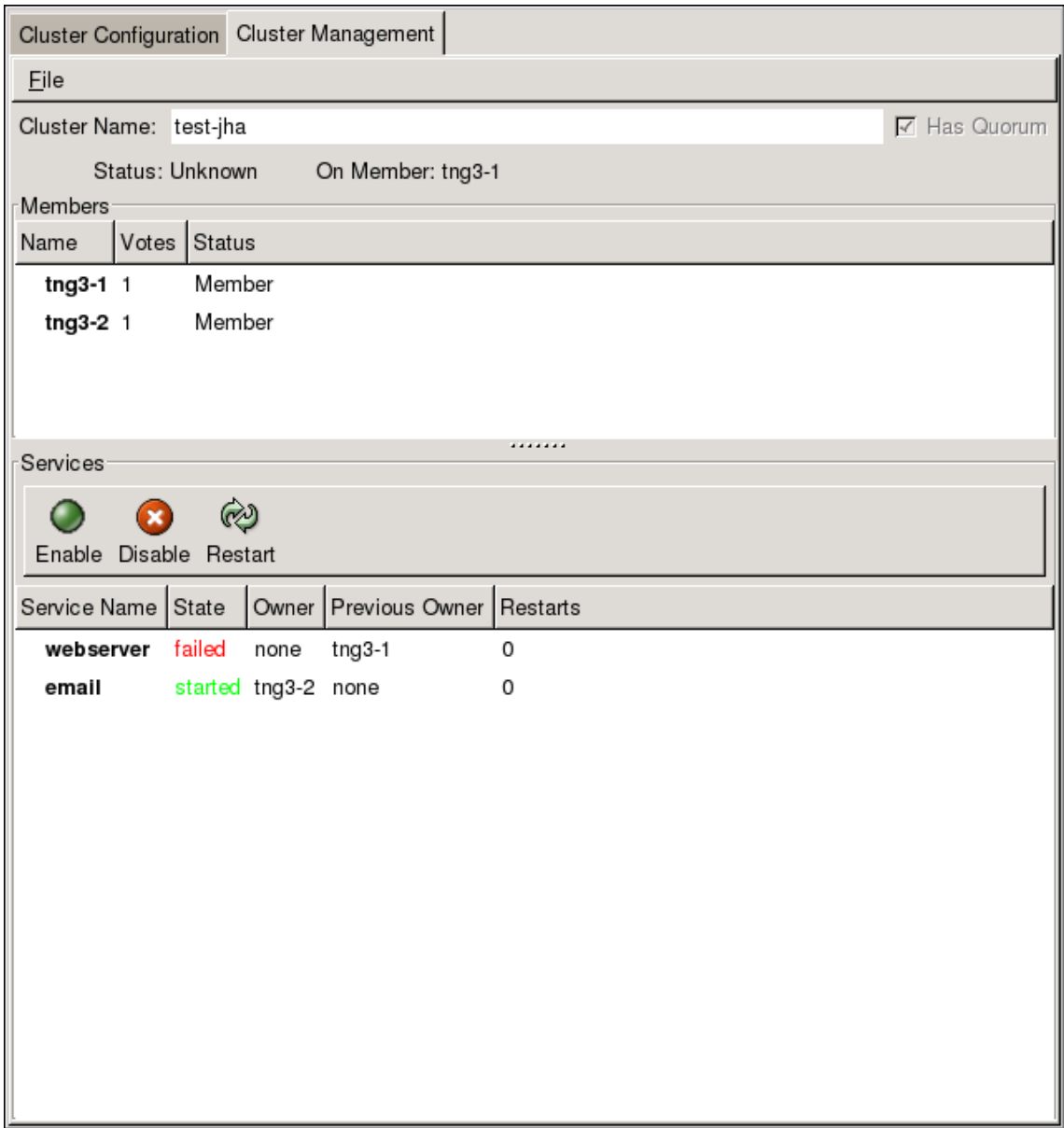


그림 1.28. Cluster Status Tool

Cluster Status Tool에서 볼 수 있는 노드 및 서비스는 클러스터 설정 파일 (/etc/cluster/cluster.conf)에 의해 결정됩니다. Cluster Status Tool를 사용하여고가용성 서비스를 활성화, 비활성화, 재시작, 재배치할 수 있습니다.

1.9.3. 명령행 관리 도구

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. 표 1.1. “명령행 도구” summarizes the command line tools.

표 1.1. 명령행 도구

명령행 도구	용도	목적
--------	----	----

명령행 도구	용도	목적
ccs_tool – 클러스터 설정 시스템 도구	Cluster Infrastructure	ccs_tool 은 클러스터 설정 파일에 온라인 업데이트를 하기 위한 프로그램입니다. 이는 클러스터 인프라 구성요소를 생성하고 편집하는 기능을 제공합니다. (예: 클러스터 생성, 노드 추가 및 삭제) 이러한 도구에 관한 자세한 내용은 ccs_tool(8) 맨 페이지를 참조하시기 바랍니다.
cman_tool – 클러스터 관리 도구	Cluster Infrastructure	cman_tool 은 CMAN 클러스터 관리자를 관리하기 위한 프로그램입니다. 이는 클러스터 연결, 클러스터 분리, 노드 삭제, 클러스터에 있는 노드의 예상 정족수 투표를 변경하는 기능을 제공합니다. 이러한 도구에 관한 자세한 내용은 cman_tool(8) 맨 페이지에서 참조하시기 바랍니다.
fence_tool – Fence 도구	Cluster Infrastructure	fence_tool 기본값으로된 fence 도메인 연결 또는 연결 해제에 사용되는 프로그램입니다. 특히, 이는 fence 데몬 (fenced)을 시작하여 도메인을 연결하고 fenced 를 종료하여 도메인에서 연결 해제 합니다. 이 도구에 관한 자세한 내용은 fence_tool(8) 맨 페이지를 참조하시기 바랍니다.
clustat – 클러스터 상태 유틸리티	고가용성 서비스 관리 구성 요소	clustat 명령은 클러스터의 상태를 보여줍니다. 이는 멤버십 정보, 정족수 보기, 클러스터된 모든 사용자 서비스 상태를 보여줍니다. 이 도구에 관한 보다 자세한 내용은 clustat(8) 맨 페이지를 참조하시기 바랍니다.
clusvcadm – 클러스터 사용자 서비스 관리 유틸리티	고가용성 서비스 관리 구성 요소	clusvcadm 명령은 클러스터에서 고가용성 서비스를 활성화, 비활성, 재배치, 재시작하게 합니다. 이 도구에 관한 보다 자세한 내용은 clusvcadm(8) 맨 페이지를 참조하시기 바랍니다.

1.10. LINUX 가상 서버 관리 GUI

다음 부분에서는 Red Hat Cluster Suite – **Piranha Configuration Tool**와 함께 사용할 수 있는 LVS 설정 도구에 대한 개요를 다루고 있습니다. **Piranha Configuration Tool**는 웹 브라우저 그래픽 사용자 인터페이스 (GUI)로 LVS에 대한 설정 파일 생성에 구조적인 접근 방법을 제공합니다 – **/etc/sysconfig/ha/lvs.cf**.

Piranha Configuration Tool를 액세스하려면 활성화된 LVS 라우터에서 **piranha-gui** 서비스가 실행 중이어야 합니다. 웹 브라우저로 **Piranha Configuration Tool**를 로컬이나 원격으로 액세스할 수 있습니다. 로컬로 액세스하려면 URL: **http://localhost:3636**을 사용하고 원격으로 액세스하려면 호스트명이나 실제 IP 주소 뒤에 **:3636**을 더합니다. 원격으로 **Piranha Configuration Tool**를 액세스하려면 **root**로 활성화된 LVS 라우터에 **ssh** 연결을 해야 합니다.

Starting the **Piranha Configuration Tool** causes the **Piranha Configuration Tool** welcome page to be displayed (refer to 그림 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.

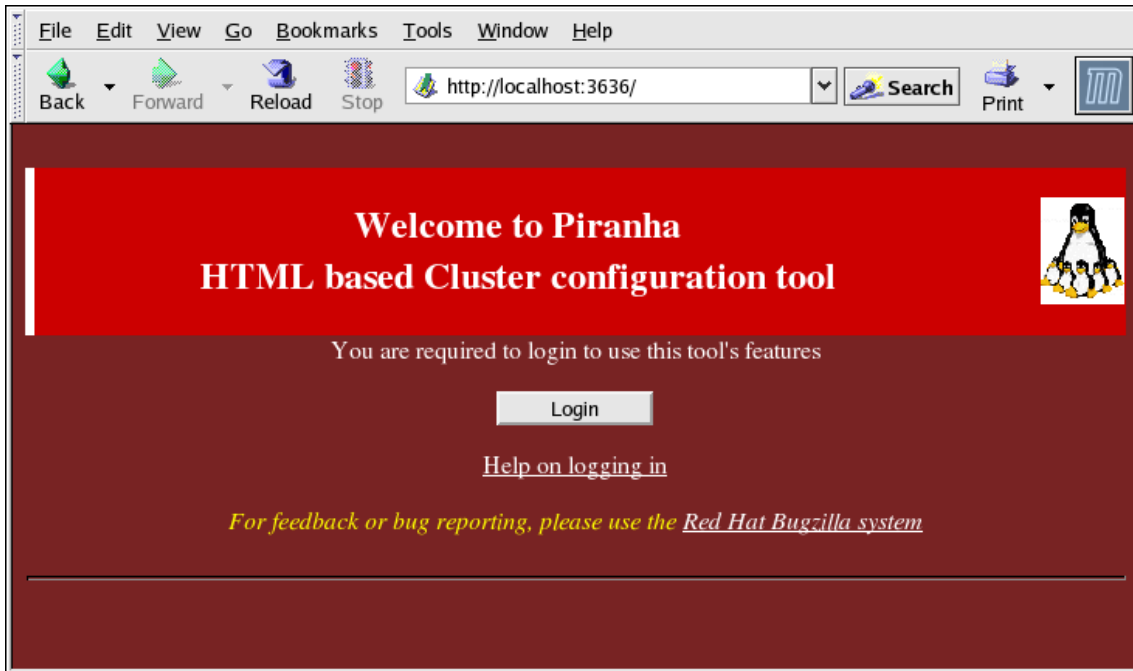


그림 1.29. The Welcome Panel

다음 부분에서는 **Piranha Configuration Tool** 설정 페이지에 대해 간략하게 설명합니다.

1.10.1. CONTROL/MONITORING

CONTROL/MONITORING 패널에서는 런타임 상태를 보여줍니다. 이는 **pulse** 데몬, LVS 라우팅 테이블, LVS 스폰 **nanny** 프로세스의 상태를 보여줍니다.

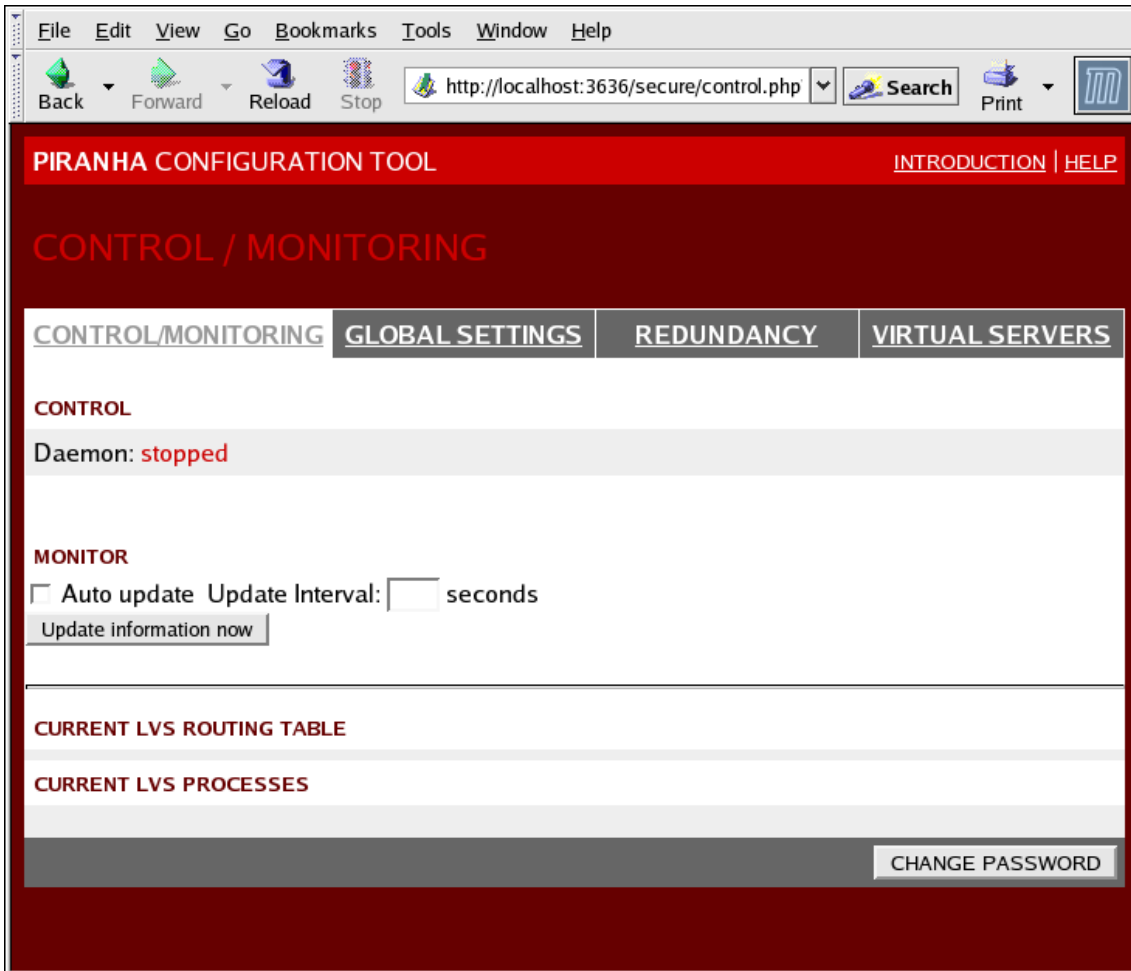


그림 1.30. The CONTROL/MONITORING Panel

Auto update

Update frequency in seconds (기본값은 10 초임) 텍스트 상자에 있는 사용자 설정 시간 간격에서 상태 보기가 자동으로 업데이트되도록 활성화합니다.

10초 보다 짧게 자동 업데이트 간격을 설정하는 것은 좋지 않습니다. 이는 페이지가 너무 자주 업데이트되어 **Auto update** 간격을 재설정하기를 어렵게 만들기 때문입니다. 이러한 문제가 발생했을 경우, 다른 패널을 클릭한 후 **CONTROL/MONITORING**으로 다시 돌아옵니다.

Update information now

상태 정보를 수동으로 업데이트합니다.

CHANGE PASSWORD

이 버튼을 클릭하면 **Piranha Configuration Tool**의 관리 암호를 변경하는 방법에 관한 내용이 있는 도움말 화면으로 이동합니다.

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.

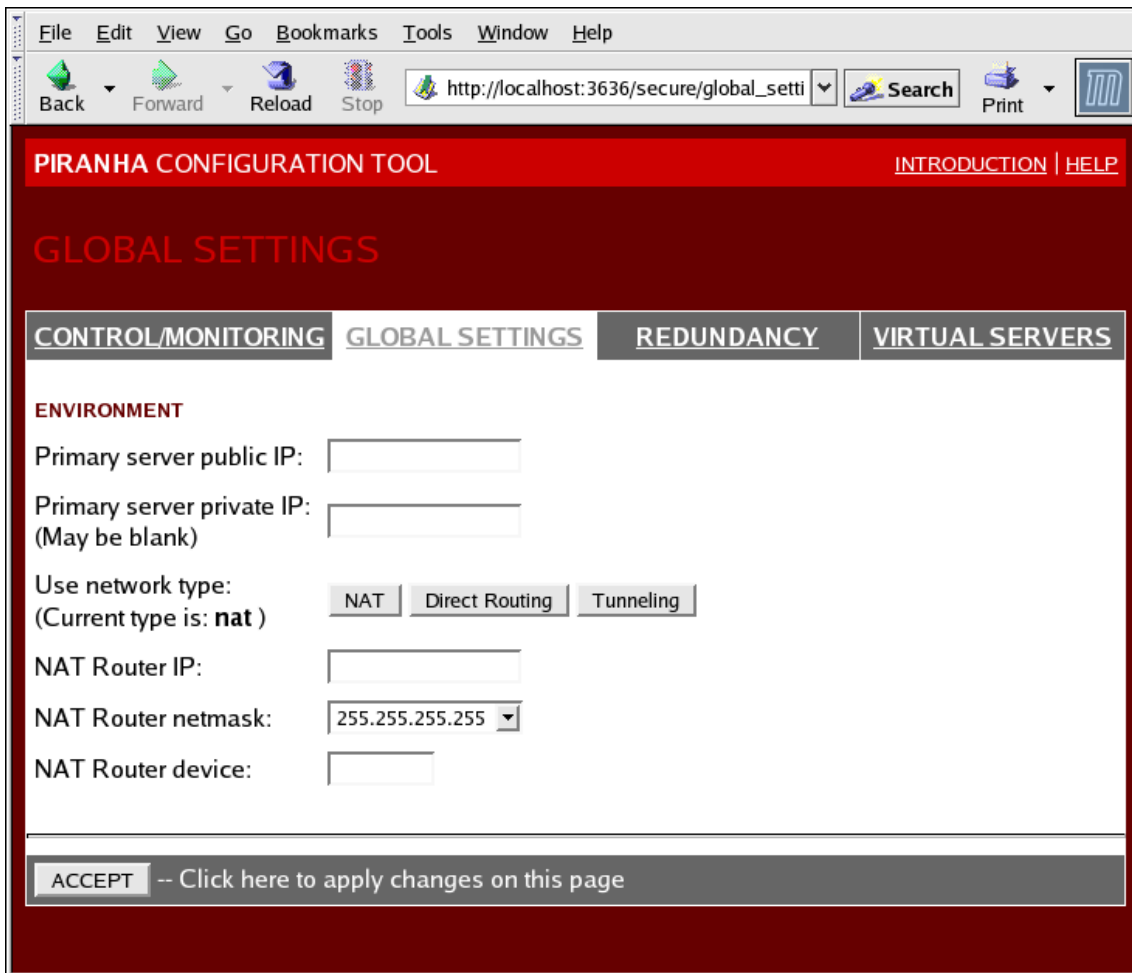


그림 1.31. The GLOBAL SETTINGS Panel

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

Primary server public IP

주요 LVS 노드에 해당하는 공개적으로 라우터 가능한 실제 IP 주소입니다.

Primary server private IP

주요 LVS 노드에 있는 다른 네트워크 인터페이스의 실제 IP 주소입니다. 이 주소는 백업 라우터의 다른 하트비트 채널로서 단독으로 사용됩니다.

Use network type

NAT 라우팅을 선택합니다.

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

개인 유동 IP를 입력합니다. 유동 IP는 실제 서버에 대해 게이트웨이로 사용되어야 합니다.

NAT Router netmask

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

NAT Router device

`eth1:1`과 같이 유동 IP 주소에 해당하는 네트워크 인터페이스의 장치 이름을 정의합니다.

1.10.3. REDUNDANCY

REDUNDANCY 패널에서는 백업 LVS 라우터 노드를 설정하고 다양한 하트비트 모니터링 옵션을 설정합니다.

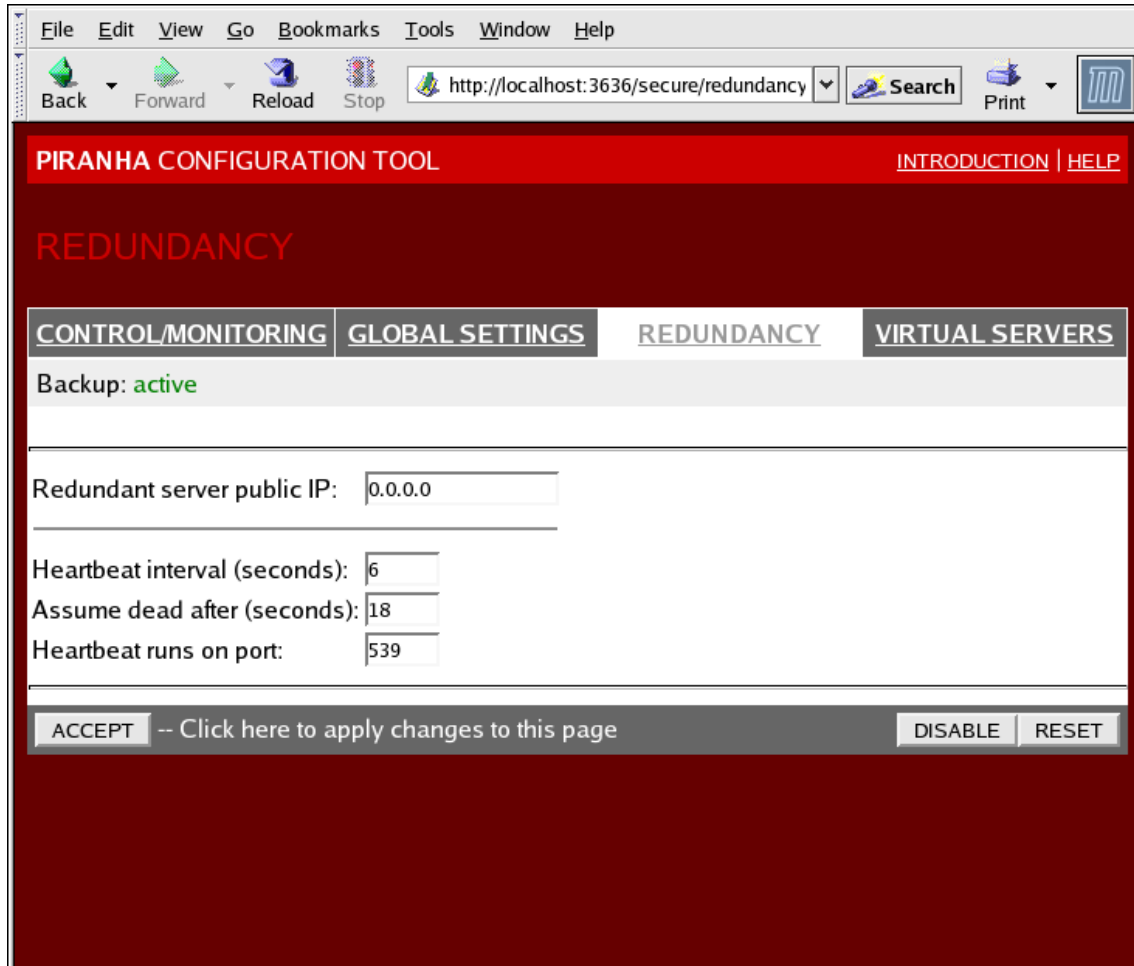


그림 1.32. The REDUNDANCY Panel

Redundant server public IP

백업 LVS 라우터 용 공개 실제 IP 주소

Redundant server private IP

The backup router's private real IP address.

나머지 패널에서는 하트비트 채널을 설정하며, 문제 발생의 경우 백업 노드가 주요 노드를 모니터링하는데 사용됩니다.

Heartbeat Interval (seconds)

하트비트 사이의 초 간격을 설정합니다 – 백업 노드가 주요 LVS 노드의 기능 상태를 점검하는 간격

Assume dead after (seconds)

몇 초가 경과한 후 주요 LVS 노드가 응답하지 않을 경우, 백업 LVS 라우터 노드가 장애 조치를 시작하게 됩니다.

Heartbeat runs on port

하트비트가 주요 LVS 노드와 통신하는 포트를 설정합니다. 빈 칸으로 되어 있을 경우 기본값으로 539를 설정합니다.

1.10.4. VIRTUAL SERVERS

VIRTUAL SERVERS 패널에서는 최근에 지정된 가상 서버에 대한 정보를 보여줍니다. 각각의 테이블 엔트리에서는 가상 서버의 상태, 서버 이름, 서버에 지정된 가상 IP, 가상 IP의 넷마스크, 서비스가 통신하는 포트 번호, 사용되는 프로토콜, 가상 장치 인터페이스의 상태를 보여 줍니다.

PIRANHA CONFIGURATION TOOL INTRODUCTION | HELP

VIRTUAL SERVERS

CONTROL/MONITORING	GLOBAL SETTINGS	REDUNDANCY	VIRTUAL SERVERS				
	STATUS	NAME	VIP	NETMASK	PORT	PROTOCOL	INTERFACE
<input type="radio"/>	up	HTTP	192.168.1.10	255.255.255.0	80	tcp	eth0:1
<input type="radio"/>	up	FTP	192.168.1.11	255.255.255.0	21	tcp	eth0:1

ADD DELETE EDIT (DE)ACTIVATE

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

그림 1.33. The **VIRTUAL SERVERS** Panel

VIRTUAL SERVERS 패널에 있는 각각의 서버는 다음의 스크린이나 또는 *하부 섹션*에서 설정할 수 있습니다.

서비스를 추가하려면 **ADD** 버튼을 클릭합니다. 서비스를 삭제하려면 가상 서버 옆에 위치한 라디오 버튼을 클릭하여 삭제하려는 서비스를 선택한 후 **DELETE** 버튼을 클릭합니다.

테이블에 있는 가상 서버를 활성화 또는 비활성화하기 위해 라디오 버튼을 클릭한 후 **(DE)ACTIVATE** 버튼을 클릭합니다.

가상 서버를 추가한 후에 왼쪽에 있는 라디오 버튼을 클릭하여 이를 설정할 수 있으며 **EDIT** 버튼을 클릭하여 **VIRTUAL SERVER** 하부 섹션을 볼 수 있습니다.

1.10.4.1. VIRTUAL SERVER 하부 섹션

The **VIRTUAL SERVER** subsection panel shown in [그림 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.

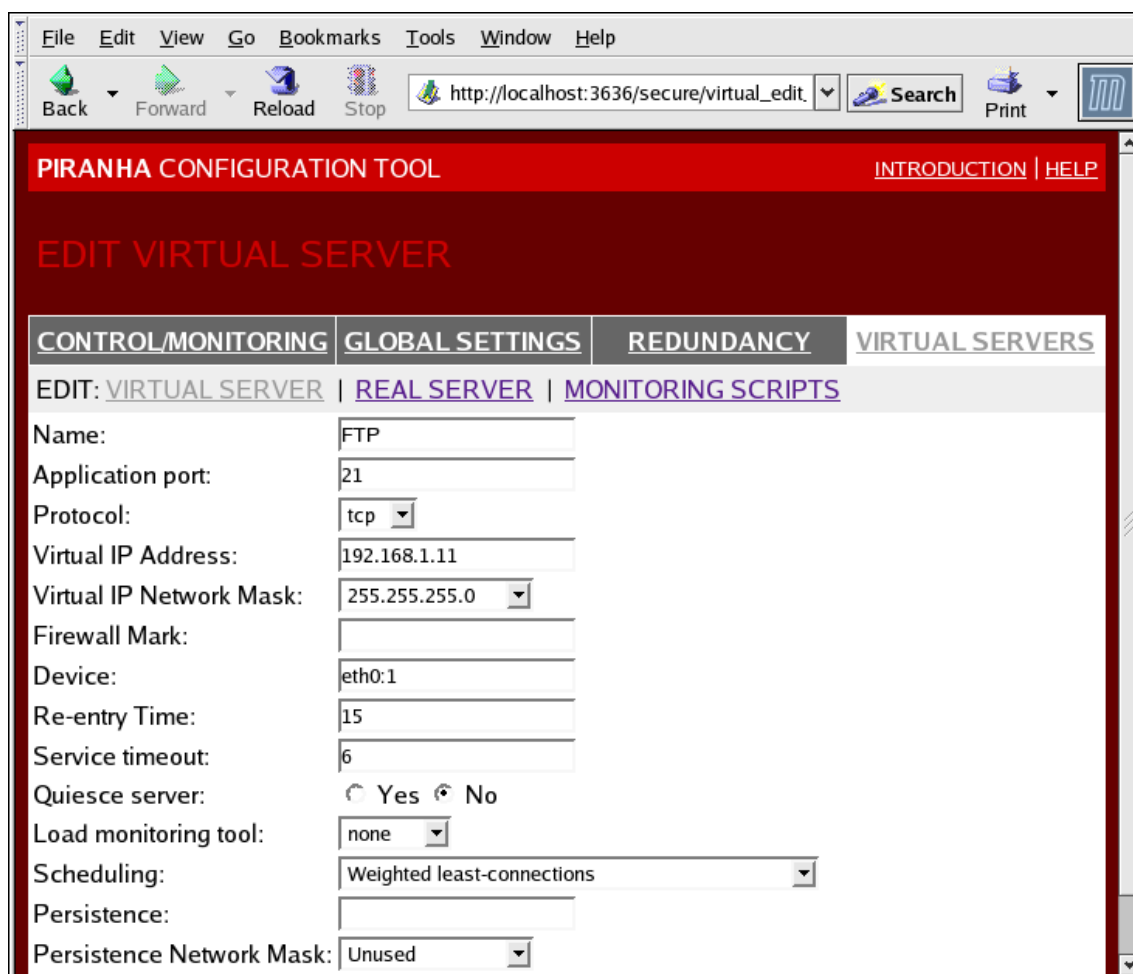


그림 1.34. The **VIRTUAL SERVERS** Subsection

Name

가상 서버를 확인하기 위한 기술적인 이름. 이는 컴퓨터의 호스트명이 *아니므로* 기술적이고 식별하기 쉬운 것으로 정해야 합니다. HTTP와 같이 가상 서버로 사용된 프로토콜을 참조할 수 있습니다.

Application port

서비스 어플리케이션이 청취할 포트 번호

Protocol

드롭 다운 메뉴에서 UDP 또는 TCP를 선택

Virtual IP Address

The virtual server's floating IP address.

Virtual IP Network Mask

드롭 다운 메뉴에 있는 가상 서버의 넷 마스크

Firewall Mark

프로토콜과 관련하여 다중 포트 프로토콜을 처리하거나 다중 포트 가상 서버를 분리하여 생성할 때 방화벽 표시 기능 정수값을 입력

Device

바인드할 **Virtual IP Address**란에 지정하고자 하는 유동 IP주소에서 네트워크 장치 이름
공개적 유동 IP 주소를 공개 네트워크에 연결된 이더넷 인터페이스에 알리아스해야 합니다.

Re-entry Time

실제 서버 연결에 실패한 후 활성 LVS 라우터가 실제 서버를 사용하기 전 초 단위로 지정된 정수 값

Service Timeout

실제 서버가 종료되었거나 사용 불가능하다고 판단하는 정수값으로된 초단위 시간

Quiesce server

Quiesce server 라디오 버튼이 선택되었을 경우, 아무때나 새로운 실제 서버 노드가 온라인이 될 수 있으며, 최소 연결 테이블은 0으로 재설정되어 활성화된 LVS 라우터는 모든 실제서버가 새로 클러스터에 추가된 것처럼 요청을 보냅니다. 이러한 옵션은 새로운 서버가 클러스터로 들어오는 최대 연결로 인해 교착 상태에 빠지지 않게 합니다.

Load monitoring tool

LVS 라우터는 **rup** 또는 **ruptime**을 사용하여 다양한 실제 서버 상의 로드를 모니터할 수 있습니다. 드롭 다운 메뉴에서 **rup**를 선택했을 경우, 각각의 실제 서버는 **rstatd** 서비스를 실행해야만 합니다. **ruptime**을 선택했을 경우, 각각의 실제 서버는 **rwhod** 서비스를 실행해야 합니다.

Scheduling

드롭 다운 메뉴에서 선호하는 스케줄링 알고리즘. 기본값은 **Weighted least-connection**으로 되어 있습니다.

Persistence

클라이언트 트랜잭션 동안 가상 서버에 지속적 연결이 필요할 경우 사용합니다. 텍스트 란에 있는 연결 제한 시간 전에 경과 시간으로 허용된 비활성화 시간(초 단위)을 지정합니다.

Persistence Network Mask

특정 서브넷에 지속성을 제한하기 위해 드롭다운 메뉴에서 적절한 네트워크 마스크를 선택합니다.

1.10.4.2. REAL SERVER 하부 섹션

패널의 상단에 있는 **REAL SERVER** 하부 섹션 링크를 클릭하면 **EDIT REAL SERVER** 하부 섹션이 나타납니다. 이는 특정 가상 서비스에 대한 물리적 서버 호스트의 상태를 보여줍니다.

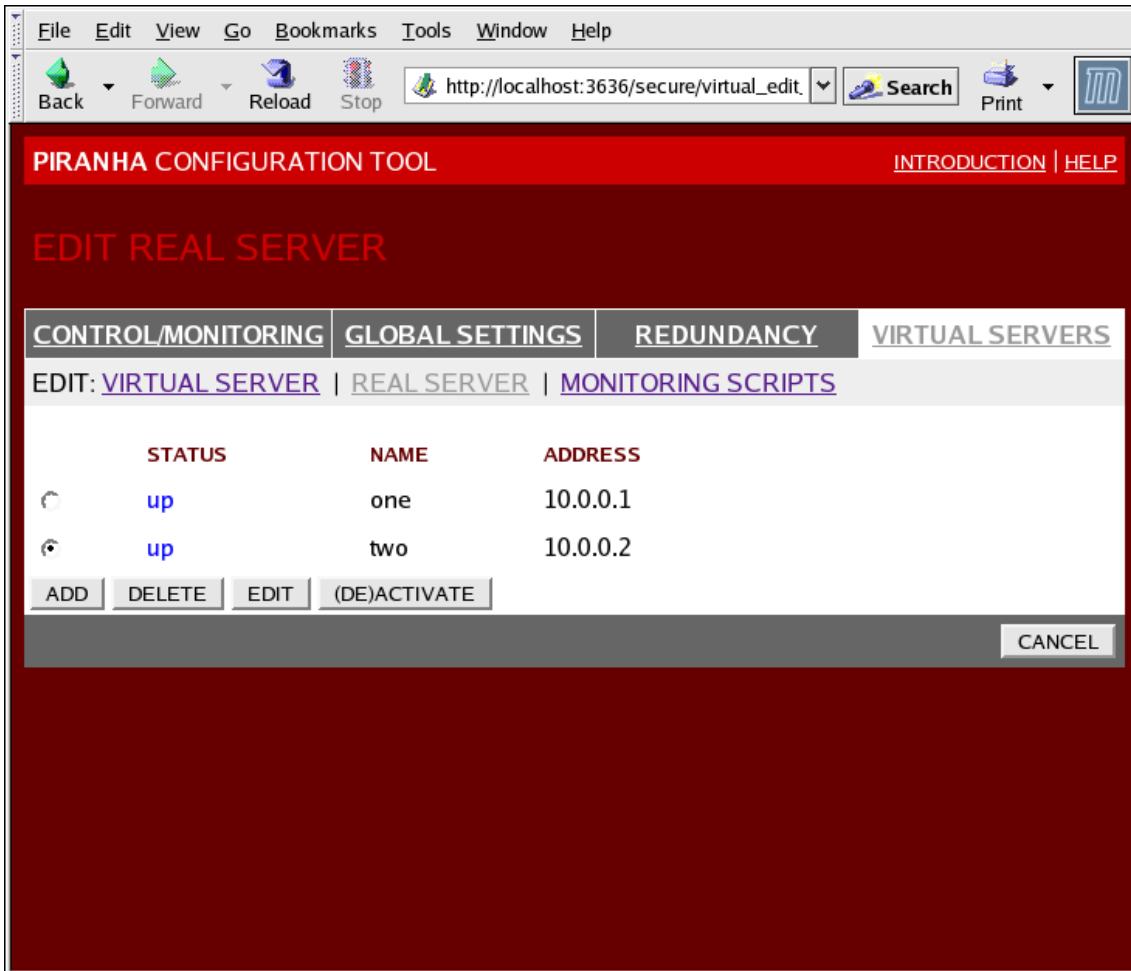


그림 1.35. The REAL SERVER Subsection

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 그림 1.36. “The REAL SERVER Configuration Panel”.

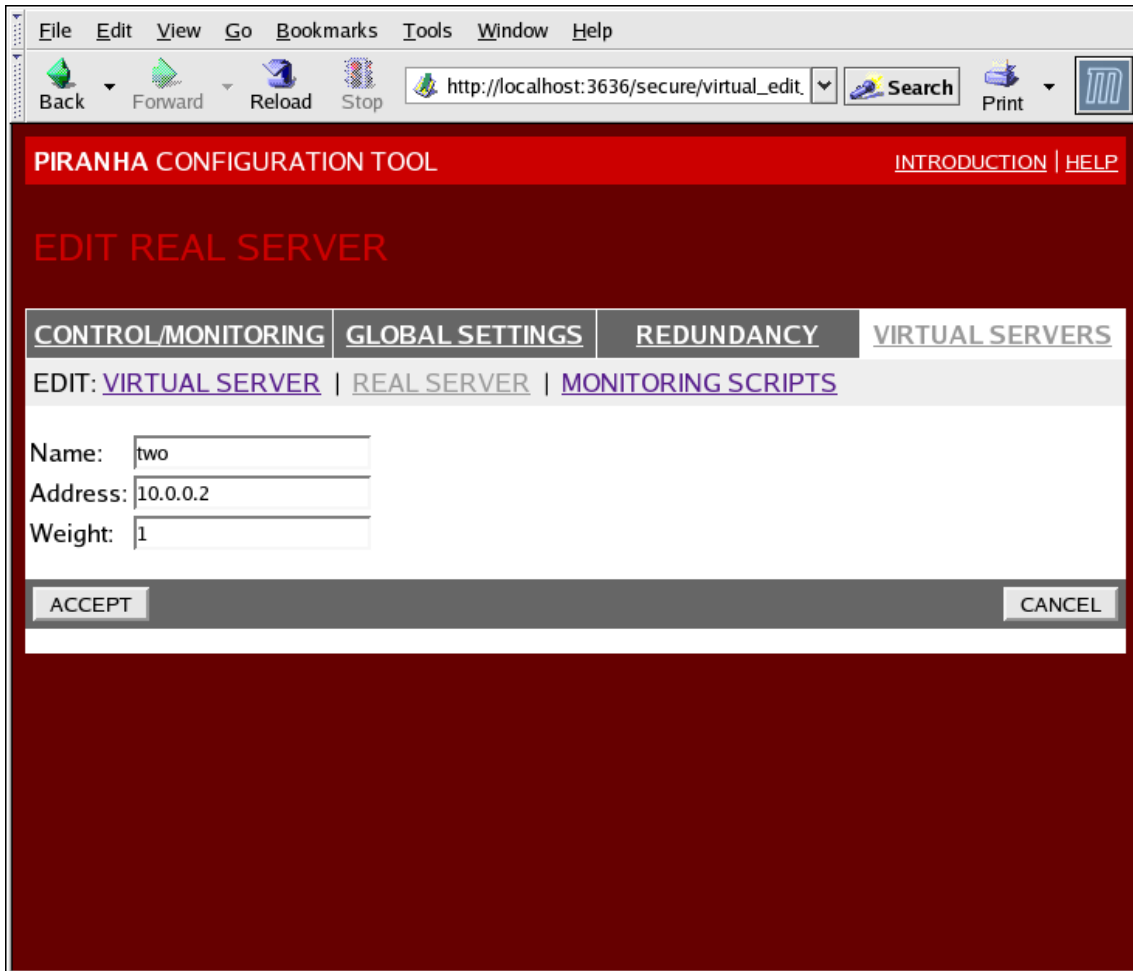
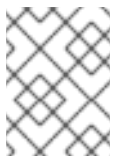


그림 1.36. The REAL SERVER Configuration Panel

이 패널은 세 가지 항목으로 구성되어 있습니다:

Name

실제 서버에 해당하는 기술적인 이름



참고

이 이름은 컴퓨터의 호스터명이 *아니므로* 기술적이고 쉽게 식별할 수 있는 것이어야 합니다.

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

페이지 상단의 **MONITORING SCRIPTS** 링크를 클릭합니다. **EDIT MONITORING SCRIPTS** 하부 섹션에서 관리자는 가상 서버의 서비스가 각각의 실제 서버에서 잘 작동하는 지를 확인하기 위해 **send/expect**

문자열 순서를 정할 수 있습니다. 또한 여기서 관리자는 역동적인 데이터 변경을 필요로 하는 서비스를 확인하기 위해 사용자 정의된 스크립트를 지정할 수 있습니다.

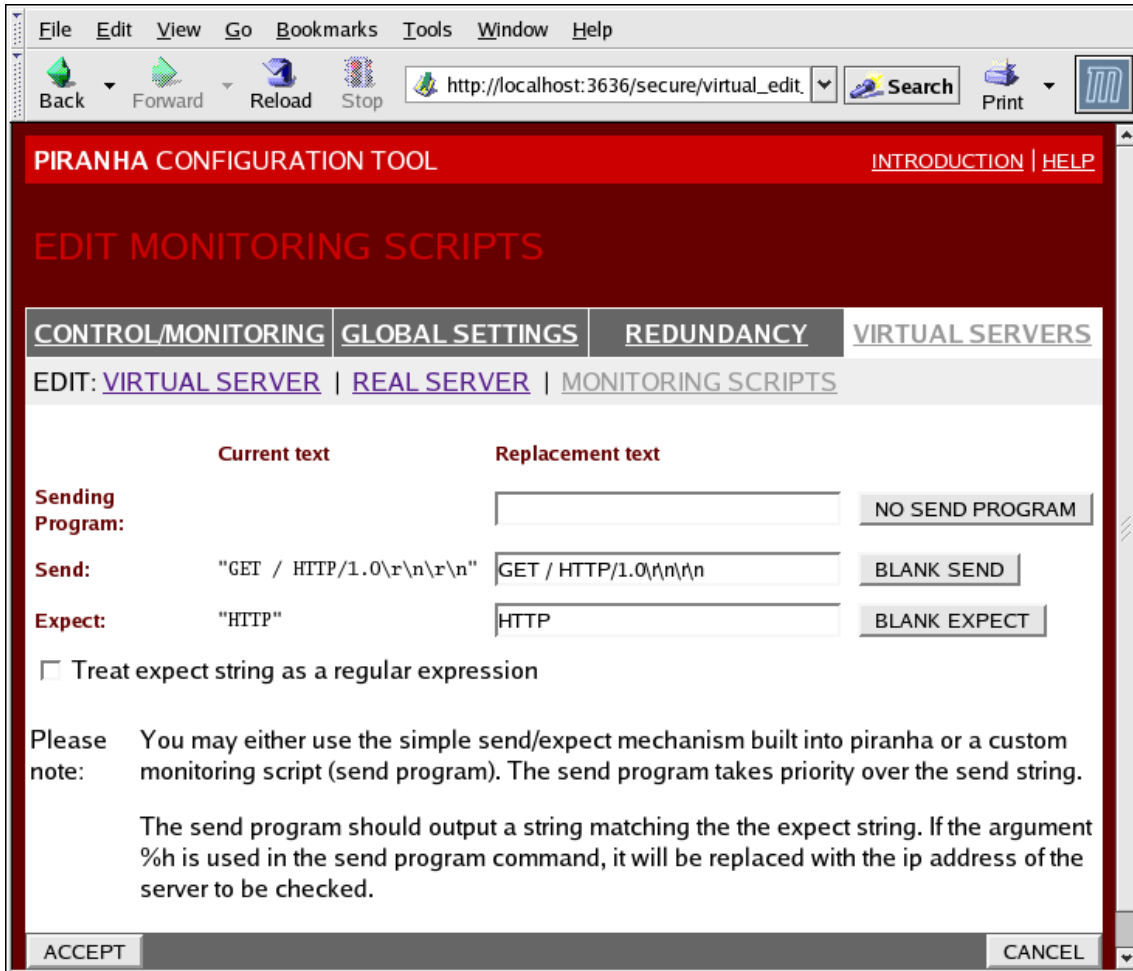
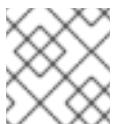


그림 1.37. The EDIT MONITORING SCRIPTS Subsection

Sending Program

보다 고급 서비스 인증을 위해 이 란에 서비스 확인 스크립트의 경로를 지정할 수 있습니다. 특히 이러한 기능은 HTTPS 또는 SSL과 같이 역동적 데이터 변경을 필요로 하는 서비스에 유용합니다.

이 기능을 사용하려면 스크립트를 작성해야 합니다. 이는 텍스트 응답 방식으로 복귀하고 실행 가능하게 설정되어 있어야 하며 **Sending Program** 란에 경로가 입력되어 있어야 합니다.



참고

외부 프로그램이 **Sending Program** 란에 입력되어 있을 경우 **Send** 란을 무시합니다.

Send

이 란에 있는 각각의 실제 서버로 보내기 위해 **nanny** 데몬에 필요한 문자열입니다. 기본적으로 이 란은 HTTP로 되어 있습니다. 필요에 따라 이 값을 변경할 수 있습니다. 이 영역을 빈 칸으로 비워두었을 경우, **nanny** 데몬은 포트를 열어 두어 서비스가 실행되고 있다고 여깁니다.

이 란에는 하나의 전송 순서만이 허용되며 인쇄할 수 있는 ASCII 문자와 다음과 같은 이스케이프 문자이어야만 합니다:

- 새 줄 표시 경우 \n

- 캐리지 리턴 경우 \r
- 탭 경우 \t
- 그 다음으로 오는 문자를 생략할 경우 \

Expect

서버가 올바르게 작동할 경우 텍스트 응답 방식으로 복귀해야 합니다. 독자적인 전송 프로그램을 사용할 경우, 전송 명령이 성공적으로 실행되었을 경우 이에 대한 응답을 입력합니다.

[1] 가상 서버는 특정 가상 IP에서 사용할 수 있도록 설정된 서비스입니다.

2장. RED HAT CLUSTER SUITE 구성 요소 요약

다음 부분에서는 Red Hat Cluster Suite 구성 요소에 대한 요약을 다루고 있으며 이는 다음과 같은 섹션으로 되어 있습니다:

- 2.1절. “클러스터 구성 요소”
- 2.2절. “맨 페이지”
- 2.3절. “하드웨어 호환성”

2.1. 클러스터 구성 요소

표 2.1. “Red Hat Cluster Suite 소프트웨어 하부 시스템 구성 요소” summarizes Red Hat Cluster Suite components.

표 2.1. Red Hat Cluster Suite 소프트웨어 하부 시스템 구성 요소

기능	구성 요소	설명
Conga	luci	원격 관리 시스템 - 관리 스테이션
	ricci	원격 관리 시스템 - 관리화된 스테이션
Cluster Configuration Tool	system-config-cluster	그래픽 설정에서 클러스터 설정 관리를 위해 사용되는 명령
CLVM (Cluster Logical Volume Manager)	clvmd	클러스터에서 LVM 메타데이터 업데이트를 사용하는 데몬. 이는 클러스터에 있는 모든 노드에서 실행되어야 하며 노드에서 이 데몬이 실행되고 있지 않을 경우 오류가 발생합니다.
	lvm	LVM2 도구. LVM2 용 명령행 도구 제공
	system-config-lvm	LVM2 용 그래픽 사용자 인터페이스 제공
	lvm.conf	LVM 설정 파일. 완전 경로는 /etc/lvm/lvm.conf . 임.
CCS (Cluster Configuration System)	ccs_tool	ccs_tool 은 CCS (Cluster Configuration System)의 일부분으로 CCS 설정 파일의 온라인 업데이트에 사용됨. 또한 GFS 6.0 (및 이전 버전)으로 생성된 CCS 아카이브에서 이번 Red Hat Cluster Suite 릴리즈를 사용한 XML 포맷 설정으로 클러스터 설정 파일을 업그레이드하는데 사용될 수 있음.
	ccs_test	ccsd 를 통해 설정 파일 정보 검색에 사용되는 진단 및 테스트 명령

기능	구성 요소	설명
	ccsd	모든 클러스터 노드에서 실행되고 클러스터 소프트웨어에 설정 파일 데이터를 제공하는 CCS 데몬
	cluster.conf	클러스터 설정 파일. 완전 경로는 /etc/cluster/cluster.conf 임
CMAN (Cluster Manager)	cman.ko	CMAN 용 커널 모듈
	cman_tool	CMAN에서의 관리용 프론트 엔드. CMAN을 시작하고 및 정지시키며 votes 와 같은 내부적 매개 변수를 변경할 수 있음
	dlm_controld	커널에 있는 dlm 을 관리하기 위해 cman init 스크립트로 시작되는 데몬; 사용자에게 의해 사용되지 않음
	gfs_controld	커널에 있는 gfs 를 관리하기 위해 cman init 스크립트로 시작되는 데몬; 사용자에게 의해 사용되지 않음.
	group_tool	fencing, DLM, GFS, 디버그 정보와 관련된 그룹 목록을 얻기 위해 사용됨; RHEL 4에서 어떤 cman_tool services가 제공되는지에 관한 정보도 얻을 수 있음
	groupd	openais/cman과 dlm_controld/gfs_controld/fenced 사이의 인터페이스에서 cman init 스크립트에 의해 시작되는 데몬; 사용자에게 의해 사용되지 않음
	libcman.so.<version number>	cman.ko 와의 통신에 필요한 프로그램 용 라이브러리
rgmanager (Resource Group Manager)	clusvcadm	클러스터에서 수동으로 사용자 서비스를 활성화, 비활성화, 재배포, 재시작하는데 사용되는 명령
	clustat	노드 멤버십 및 서버스 실행을 포함한 클러스터의 상태를 나타내는데 사용되는 명령
	clurgmgrd	서비스 시작, 서비스 비활성화, 서비스 재배포, 서비스 재시작을 포함하여 사용자 서비스 요청을 처리하기 위해 사용되는 데몬

기능	구성 요소	설명
	clurmtabd	클러스터된 NFS 마운트 테이블을 처리하기 위해 사용되는 데몬
Fence	fence_apc	APC 전원 스위치 용 Fence 에이전트
	fence_bladecenter	Telnet 인터페이스를 사용하는 IBM Bladecenters 용 Fence 에이전트
	fence_bullpap	Bull Novascale PAP (Platform Administration Processor) 인터페이스 용 Fence 에이전트
	fence_drac	DRAC (Dell Remote Access Card) 용 Fencing 에이전트
	fence_ipmilan	LAN을 통해 IPMI (Intelligent Platform Management Interface)에 의해 관리되는 컴퓨터에 해당하는 Fence 에이전트
	fence_wti	WTI 전원 스위치 용 Fence 에이전트
	fence_brocade	Brocade Fibre 채널 스위치 용 Fence 에이전트
	fence_mcddata	McData Fibre 채널 스위치 용 Fence 에이전트
	fence_vixel	Vixel Fibre 채널 스위치 용 Fence 에이전트
	fence_sanbox2	SANBox2 Fibre 채널 스위치 용 Fence 에이전트
	fence_ilo	HP ILO 인터페이스 (이전에는 fence_rib) 용 Fence 에이전트
	fence_rsa	IBM RSA II 용 I/O Fencing 에이전트
	fence_gnbd	GNBD 스토리지와 함께 사용되는 Fence 에이전트
	fence_scsi	SCSI 영구 예약 용 I/O fencing 에이전트
	fence_egera	Egenera BladeFrame 시스템과 함께 사용되는 Fence 에이전트
	fence_manual	수동 인터렉션 용 Fence 에이전트. <i>알림</i> 이는 제품 용으로 지원되지 않음

기능	구성 요소	설명
	fence_ack_manual	fence_manual 에이전트 용 사용자 인터페이스
	fence_node	단일 노드에서 I/O fencing을 실행하는 프로그램
	fence_xvm	Xen 가상 머신 용 I/O Fencing 에이전트
	fence_xvmd	Xen 가상 머신 용 I/O Fencing 에이전트 호스트
	fence_tool	fence 데몬 연결 및 연결 해제 프로그램
	fenced	I/O Fencing 데몬
DLM	libdlm.so.<version number>	DLM (Distributed Lock Manager) 지원용 라이브러리
GFS	gfs.ko	GFS 파일 시스템을 실행하고 GFS 클러스터 노드로 읽어오는 커널 모듈
	gfs_fsck	마운트 해제된 GFS 파일 시스템 복구 명령
	gfs_grow	마운트된 GFS 파일 시스템을 증가시키는 명령
	gfs_jadd	마운트된 GFS 파일 시스템 저널 추가 명령
	gfs_mkfs	스토리지에 GFS 파일 시스템 생성 명령
	gfs_quota	마운트된 GFS 파일 시스템 쿼터 관리 명령
	gfs_tool	GFS 파일 시스템을 설정 또는 조정 명령. 이 명령으로 파일 시스템에 관한 다양한 정보를 얻을 수 있음
	mount.gfs	mount (8) 로 볼 수 있는 마운트에 관한 도움말; 사용자에게 의해 사용되지 않음
GNBD	gnbd.ko	클라이언트에서 GNBD 장치 드라이버를 실행하는 커널 모듈.
	gnbd_export	GNBD 서버에서 GNBD를 생성하고 내보내며 관리하기 위한 명령

기능	구성 요소	설명
	gnbd_import	GNBD 클라이언트에서 GNBD를 관리하고 읽어오기 위한 명령
	gnbd_serv	네트워크를 통해 로컬 스토리지를 내보내기 위해 노드를 허용하는 서버 데몬
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 <code>/etc/rc.d/init.d/pulse</code> script. It then reads the configuration file <code>/etc/sysconfig/ha/lvs.cf</code> . 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	lvs 데몬이 일단 pulse 에 의해 호출되면 활성화된 LVS 라우터에서 실행됩니다. 이는 <code>/etc/sysconfig/ha/lvs.cf</code> 설정 파일을 읽으며, ipvsadm 유틸리티를 호출하여 IPVS 라우팅 테이블을 생성하고 관리하며 각 각의 설정된 LVS 서비스 용 nanny 프로세스를 지정합니다. nanny 가 서버가 다운되었다고 보고하면 lvs 가 ipvsadm 유틸리티를 지시하여 IPVS 라우팅 테이블에서 서버를 삭제합니다.
	ipvsadm	커널에서 IPVS 라우팅 테이블을 업데이트함. lvs 데몬은 IPVS 라우팅 테이블에 있는 항목을 추가, 변경, 삭제하기 위해 ipvsadm 명령을 호출하여 LVS를 설정 및 관리함.
	nanny	nanny 모니터링 데몬은 LVS 라우트에서 실행됨. 이 데몬을 통해, 활성화된 LVS 라우터는 실제 서버 상태를 결정하며, 옵션으로 작업량을 모니터링함. 분리된 프로세스는 실제 서버에 정의된 각각의 서비스에 대해 실행됨.

기능	구성 요소	설명
	lvs.cf	LVS 설정 파일로 완전 경로는 /etc/sysconfig/ha/lvs.cf 임. 직접 또는 간접적으로 모든 데몬은 이 파일에서 파일 설정 정보를 얻음.
	Piranha Configuration Tool	LVS 모니터링, 설정, 관리 용 웹 기반 도구로 /etc/sysconfig/ha/lvs.cf LVS 설정 파일을 관리하기 위한 기본값 도구.
	send_arp	이 프로그램은 장애조치 중 하나의 노드에서 다른 노드로 유동 IP 주소가 변경되었을때 ARP 브로드캐스트를 전송함.
Quorum 디스크	qdisk	CMAN / Linux-Cluster 용 디스크 기반 quorum 데몬
	mkqdisk	클러스터 Quorum 디스크 유틸리티
	qdiskd	클러스터 Quorum 디스크 데몬

2.2. 맨 페이지

다음 부분에는 추가 자료로서 Red Hat Cluster Suite와 관련된 맨 페이지의 목록이 있습니다.

- 클러스터 인프라
 - **ccs_tool (8)** - CCS 설정 파일의 온라인 업데이트를 위해 사용되는 도구
 - **ccs_test (8)** - 클러스터 설정 시스템 실행을 위한 진단 도구
 - **ccsd (8)** - CCS 클러스터 설정 파일 액세스에 사용되는 데몬
 - **ccs (7)** - 클러스터 설정 시스템
 - **cman_tool (8)** - 클러스터 관리 도구
 - **cluster.conf [cluster] (5)** - 클러스터 제품 용 설정 파일
 - **qdisk (5)** - CMAN / Linux-Cluster 용 디스크 기반 quorum 데몬
 - **mkqdisk (8)** - 클러스터 Quorum 디스크 유틸리티
 - **qdiskd (8)** - 클러스터 Quorum 디스크 데몬
 - **fence_ack_manual (8)** - 수동 I/O Fencing으로서 오퍼레이터에 의해 실행되는 프로그램
 - **fence_apc (8)** - APC MasterSwitch 용 I/O Fencing 에이전트
 - **fence_bladecenter (8)** - IBM Bladecenter 용 I/O Fencing 에이전트
 - **fence_brocade (8)** - Brocade FC 스위치 용 I/O Fencing 에이전트

- fence_bullpap (8) - PAP 관리 콘솔에 의해 제어되는 Bull FAME 아키텍처 용 I/O Fencing 에이전트
- fence_drac (8) - Dell Remote Access Card 용 fencing 에이전트
- fence_egenera (8) - Egenera BladeFrame 용 I/O Fencing 에이전트
- fence_gnbd (8) - GNBD 기반 GFS 클러스터 용 I/O Fencing 에이전트
- fence_ilo (8) - HP Integrated Lights Out Card 용 I/O Fencing 에이전트
- fence_ipmilan (8) - LAN을 통해 IPMI에 의해 제어되는 기계 장치 용 I/O Fencing 에이전트
- fence_manual (8) - 수동 I/O Fencing 부분으로 fenced에 의해 실행되는 프로그램
- fence_mcddata (8) - McData FC 스위치 용 I/O Fencing 에이전트
- fence_node (8) - 단일 노드에서 I/O fencing을 실행하는 프로그램
- fence_rib (8) - Compaq Remote Insight Lights Out Card 용 I/O Fencing 에이전트
- fence_rsa (8) - IBM RSA II 용 I/O Fencing 에이전트
- fence_sanbox2 (8) - QLogic SANBox2 FC 스위치 용 I/O Fencing 에이전트
- fence_scsi (8) - SCSI 영구 예약 용 I/O fencing 에이전트
- fence_tool (8) - fence 데몬 연결 및 연결 해제 프로그램
- fence_vixel (8) - Vixel FC 스위치 용 I/O Fencing 에이전트
- fence_wti (8) - WTI Network Power Switch 용 I/O Fencing 에이전트
- fence_xvm (8) - Xen 가상 머신 용 I/O Fencing 에이전트
- fence_xvmd (8) - Xen 가상 머신 용 I/O Fencing 에이전트
- fenced (8) - I/O Fencing 데몬
- 고가용성 서비스 관리
 - clusvcadm (8) - 클러스터 사용자 서비스 관리 유틸리티
 - clustat (8) - 클러스터 상태 유틸리티
 - Clurgmgrd [clurgmgrd] (8) - 리소스 그룹 (클러스터 서비스) 관리자 데몬
 - clurmtabd (8) - 클러스터 NFS 원격 마운트 테이블 데몬
- GFS
 - gfs_fsck (8) - 오프라인 GFS 파일 시스템 확인
 - gfs_grow (8) - GFS 파일 시스템 확장
 - gfs_jadd (8) - GFS 파일 시스템에 저널 추가
 - gfs_mount (8) - GFS 마운트 옵션

- gfs_quota (8) - GFS 디스크 쿼터 조작
- gfs_tool (8) - gfs ioctl 호출에서의 인터페이스
- CLVM (Cluster Logical Volume Manager)
 - clvmd (8) - 클러스터 LVM 데몬
 - lvm (8) - LVM2 도구
 - lvm.conf [lvm] (5) - LVM2 용 설정 파일
 - lvmchange (8) - 논리 볼륨 관리자의 속성 변경
 - pvcreate (8) - LVM 사용을 위한 디스크 또는 파티션 초기화
 - lvs (8) - 논리 볼륨에 관한 내용 보고
- GNBD (Global Network Block Device)
 - gnbd_export (8) - GNBD를 내보내기 위한 인터페이스
 - gnbd_import (8) - 클라이언트에서 GNBD 블록 장치 조작
 - gnbd_serv (8) - gnbd 서버 데몬
- LVS
 - pulse (8) - 클러스터 노드의 상태를 모니터링하기 위한 heartbeating 데몬
 - lvs.cf [lvs] (5) - lvs 용 설정 파일
 - lvscan (8) - 논리 볼륨을 위해 (모든 디스크를) 스캔
 - lvsd (8) - Red Hat 클러스터링 서비스를 제어하기 위한 데몬
 - ipvsadm (8) - Linux 가상 서버 관리
 - ipvsadm-restore (8) - stdin에서 IPVS 테이블 복구
 - ipvsadm-save (8) - stdout에 IPVS 테이블 저장
 - nanny (8) - 클러스터에 있는 서비스의 상태를 모니터링하기 위한 도구
 - send_arp (8) - 새 IP 주소 / MAC 주소 매핑을 위한 네트워크 확인 도구

2.3. 하드웨어 호환성

Red Hat Cluster Suite 구성 요소와 호환되는 하드웨어에 관한 정보는 (예: 지원되는 fence 장치, 스토리지 장치, Fibre 채널 스위치) http://www.redhat.com/cluster_suite/hardware/에 있는 하드웨어 설정 가이드를 참조하시기 바랍니다.

부록 A. 개정 내역

고침 3-8.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
고침 3-8 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
고침 1.0-0 포인트 릴리즈 통합	Tue Jan 20 2008	Paul Kennedy

색인

C

cluster

displaying status, [Cluster Status Tool](#)

cluster administration

displaying cluster and service status, [Cluster Status Tool](#)

cluster component compatible hardware, [하드웨어 호환성](#)

cluster component man pages, [맨 페이지](#)

cluster components table, [클러스터 구성 요소](#)

Cluster Configuration Tool

accessing, [Cluster Configuration Tool](#)

cluster service

displaying status, [Cluster Status Tool](#)

command line tools table, [명령행 관리 도구](#)

compatible hardware

cluster components, [하드웨어 호환성](#)

Conga

overview, [Conga](#)

Conga overview, [Conga](#)

F

feedback, [피드백](#)

I

introduction, [소개](#)

other Red Hat Enterprise Linux documents, [소개](#)

L

LVS

direct routing

requirements, hardware, [직접 라우팅](#)

requirements, network, [직접 라우팅](#)

requirements, software, [직접 라우팅](#)

routing methods

NAT, [라우팅 방식](#)

three tiered

high-availability cluster, [Three-Tier LVS Topology](#)

M

man pages

cluster components, [맨 페이지](#)

N

NAT

routing methods, LVS, [라우팅 방식](#)

network address translation ([살펴볼 내용 NAT](#))

O

overview

economy, [Red Hat GFS](#)

performance, [Red Hat GFS](#)

scalability, [Red Hat GFS](#)

P

Piranha Configuration Tool

CONTROL/MONITORING, [CONTROL/MONITORING](#)

EDIT MONITORING SCRIPTS Subsection, [EDIT MONITORING SCRIPTS Subsection](#)

GLOBAL SETTINGS, [GLOBAL SETTINGS](#)

login panel, [Linux 가상 서버 관리 GUI](#)

necessary software, [Linux 가상 서버 관리 GUI](#)

REAL SERVER subsection, [REAL SERVER 하부 섹션](#)

REDUNDANCY, [REDUNDANCY](#)

VIRTUAL SERVER subsection, [VIRTUAL SERVERS](#)

Firewall Mark , [VIRTUAL SERVER 하부 섹션](#)

Persistence , [VIRTUAL SERVER 하부 섹션](#)

Scheduling , [VIRTUAL SERVER 하부 섹션](#)

Virtual IP Address , [VIRTUAL SERVER 하부 섹션](#)

VIRTUAL SERVERS, [VIRTUAL SERVERS](#)

R

Red Hat Cluster Suite

components, [클러스터 구성 요소](#)

T

table

cluster components, [클러스터 구성 요소](#)

command line tools, [명령행 관리 도구](#)