Red Hat Enterprise Linux 6
가상 서버 관리

Red Hat Enterprise Linux 용 로드 밸런서 추가 기능
엮음 6

Landmann

Red Hat Enterprise Linux 6 가상 서버 관리

Red Hat Enterprise Linux 용 로드 밸런서 추가 기능
엮음 6

Landmann
rlandmann@redhat.com

법적 공지

초록

로드 밸런서 추가 기능 (Load Balancer Add-On) 시스템을 구축하는 것은 PIRANHA를 통해 설정되는 라우팅과 로드 밸런싱 기술을 위해 특화된 LVS (Linux Virtual Servers)를 이용해 높은 가용성과 확장성의 솔루션을 제공합니다. 이 문서에서는 Red Hat Enterprise Linux로 고성능 시스템과 서비스를 설정하는 방법과 Red Hat Enterprise Linux 6의 로드 밸런서 추가 기능(Load Balancer Add-On)에 대해 설명합니다.

차례

소개

이 문서는 로드 밸런서 추가 기능 (Load Balancer Add-On) 구성요소를 설치하고, 설정하고, 관리하는 방법에 대한 정보를 제공합니다. 로드 밸런서 추가 기능은 서버의 풀(pool)에 대해 트래픽을 배분하는 특별한 라우팅 기법을 통해 부하 균등화를 제공합니다.

이 문서는 Red Hat Enterprise Linux와 클러스터, 저장소, 서버 컴퓨팅에 대해 심도깊은 이해를 가지고 있는 사람들을 대상으로 합니다.

이 문서는 다음과 같이 구성되어 있습니다:

- 1장. *로드 밸런서 추가 기능 개요*
- 2장. *로드 밸런서 추가 기능 (Load Balancer Add-On) 초기 설정*
- 3장. *로드 밸런서 추가 기능 설정*
- 4장. ***Piranha Configuration Tool**를 사용하여 로드 밸런서 추가 기능 (Load Balancer Add-On) 설정하기*
- 부록 A. *고가용성 추가 기능과 함께 로드 밸런서 추가 기능 사용하기*

Red Hat Enterprise Linux 6에 대한 더 자세한 정보는 다음을 참조하십시오:

- *Red Hat Enterprise Linux 설치 가이드* — Red Hat Enterprise Linux 6 설치와 관련된 정보를 제공합니다.
- *Red Hat Enterprise Linux 운용가이드* — Red Hat Enterprise Linux 6의 배포, 설정, 관리에 대한 정보를 제공합니다

로드 밸런서 추가 기능 및 Red Hat Enterprise Linux 6 관련 제품에 관한 보다 자세한 내용은 다음에서 참조하시기 바랍니다:

- *Red Hat Cluster Suite 개요* — 고가용성 추가 기능, 장애 복구형 스토리지 추가 기능, 로드 밸런서 추가 기능에 대한 개요를 자세히 다루고 있습니다.
- *고가용성 추가 기능 설정 및 관리* — Red Hat Enterprise Linux 6의 고가용성 추가 기능 (Red Hat Cluster라고 알려짐)을 설정 및 관리하는 방법에 대해 다루고 있습니다.
- *Logical Volume Manager Administration* — 논리 볼륨 관리자(Logical Volume Manager,LVM)를 클러스터 환경에서 돌리는 것을 포함하여, LVM에 대한 설명을 제공합니다.
- *GFS2 (Global File System 2): 설정 및 관리* — Red Hat 장애 복구형 스토리지 추가 기능 (Red Hat GFS2라고도 알려짐)을 설치, 설정, 관리하는 방법에 대한 내용을 다루고 있습니다.
- *다중 경로 장치 매퍼 (DM Multipath)* — Red Hat Enterprise Linux 6의 다중 경로 장치 매퍼 (Device-Mapper Multipath) 사용에 관한 자세한 내용을 다루고 있습니다.
- *릴리즈 노트* — Red Hat 제품의 최신 릴리즈에 대한 내용을 제공합니다.

이 문서와 다른 Red Hat 문서들은 HTML, PDF 또는 RPM 버전으로 Red Hat Enterprise Linux 문서 CD와 온라인 http://www.redhat.com/docs/에서 찾을 수 있습니다..

1. 문서화 규정

이 메뉴얼에서는 특정 단어 및 구문을 강조 표시하여 특정 정보 부분에 주의를 집중시키기 위해 문서화 규정을 사용하고 있습니다.

PDF 및 문서 편집에서 이 메뉴얼은 Liberation 글꼴 모음에 있는 서체를 사용합니다. 시스템에 Liberation 글꼴 모음이 설치되어 있을 경우 이는 HTML 편집에서도 사용되지만 설치되어 있지 않을 경우, 다른 동일한 서체로 나타나게 됩니다. 알림: Red Hat Enterprise Linux 5 및 이후 버전에는 기본값으로 Liberation 글꼴 모음이 들어 있습니다.

1.1. 표기 규정

네 가지 표기 규정을 사용하여 특정 단어 및 구문에 주의를 집중시킵니다. 이러한 규정 및 적용 방식은 다음과 같습니다.

고정폭 굵은체

쉘 명령, 파일 이름 및 경로를 포함한 시스템 입력을 강조하기 위해 사용됩니다. 키 및 키 조합을 강조하기 위해 사용되기도 합니다. 예:

> 현재 작업 중인 디렉토리에 있는 **my_next_bestselling_novel** 파일 내용을 확인하려면, 쉘 프롬프트에서 **cat my_next_bestselling_novel** 명령을 입력하고 **Enter** 키를 눌러 명령을 실행합니다.

위에서 파일 이름, 쉘 명령, 키 모두는 고정폭 굵은체로 나타나 있어 내용과 구별될 수 있습니다.

키 조합은 키 조합의 각 부분을 더하기(+) 기호로 연결하여 개별 키과 구별되게 할 수 있습니다. 예:

> **Enter** 키를 눌러 명령을 실행합니다.
>
> **Ctrl+Alt+F2**를 눌러 가상 터미널로 전환합니다.

첫 번째 예에서는 눌러야 하는 특정 키를 강조하고 있습니다. 두 번째 예에서는 키 조합 (동시에 눌러야 하는 세 개의 키 묶음)을 강조하고 있습니다.

소스 코드를 설명해야 할 경우, 문장에서 언급된 클래스 이름, 방식, 기능, 변수 이름 및 반환값은 위와 같이 **고정폭 굵은체**로 나타나게 됩니다. 예:

> 파일 관련 클래스에는 파일 시스템의 경우 **filesystem**, 파일의 경우 **file**, 디렉토리의 경우 **dir**가 포함됩니다. 각각의 클래스에는 자체의 권한 설정이 있습니다.

가변폭 굵은체

이는 프로그램 이름; 대화 상자 텍스트; 레이블된 버튼; 체크 박스 및 라디오 버튼 레이블; 메뉴 제목; 하부 메뉴 제목을 포함하여 시스템에 있는 단어 또는 구문을 나타냅니다. 예:

> 주 메뉴 바에서 **시스템 → 기본설정 → 마우스**를 선택하여 **마우스 기본 설정**을 시작합니다. **버튼** 탭에서, **왼손 잡이 마우스** 체크 상자를 선택하고 **닫기**를 클릭하여 주요 마우스 버튼을 왼쪽에서 오른쪽으로 전환합니다 (왼손 잡이일 경우 보다 적절하게 마우스 사용을 할 수 있게 함).
>
> **gedit** 파일에 특수 문자를 입력하려면 주 메뉴 바에서 **프로그램 → 보조 프로그램 → 문자 표**를 선택합니다. 그 다음으로, **문자 표** 메뉴 바에서 **찾기 → 찾기...**를 선택하고 **찾기** 필드에 문자를 입력하고 **다음**을 클릭합니다. 찾기한 문자가 **문자 표**에 강조 표시되어 나타납니다. 강조 표시된 문자를 두 번 클릭하여 **복사할 텍스트** 필드에 나타나면 **복사** 버튼을 클릭합니다. 편집 중인 문서로 돌아가 **gedit** 메뉴 바에서 **편집 → 붙여넣기**를 선택합니다.

위의 내용에는 프로그램 이름, 다양한 시스템 메뉴 이름 및 항목; 특정 프로그램 메뉴 이름; GUI 인터페이스에 있는 버튼 및 텍스트가 포함되어 있으며, 텍스트와 구별 가능하도록 모두 가변폭 굵은체로 되어 있습니다.

고정폭 굵은 이텔릭체 또는 ***가변폭 굵은 이텔릭체***

고정폭 굵은체이던 가변폭 굵은체이던지 간에 이텔릭체가 추가될 경우 이는 교체 또는 변경 가능한 텍스트를 나타내는 것입니다. 글자 그대로 입력하지 말아야 할 텍스트나 또는 상황에 따라 변경해야 하는 텍스트의 경우 이텔릭체로 나타냅니다. 예:

> ssh를 사용하여 원격 컴퓨터에 연결하려면, 쉘 프롬프트에 **ssh *username@domain.name***을

입력합니다. 원격 컴퓨터가 **example.com**이고 사용자 이름이 john일 경우, **ssh john@example.com**을 입력합니다.

mount -o remount *file-system* 명령은 지정한 파일 시스템을 다시 마운트합니다. 예를 들어, **/home** 파일 시스템을 다시 마운트하려면 **mount -o remount /home** 명령을 사용합니다.

현재 설치된 패키지 버전을 보려면, **rpm -q *package*** 명령을 사용합니다. 그러면 다음과 같은 값이 출력됩니다: ***package-version-release***.

사용자 이름, 도메인 이름, 파일 시스템, 패키지, 버전 및 릴리즈는 굵은 이탤릭체로 표시되는 점에 유의하십시오. 각 단어는 자리 표시자로 명령을 실행할 때 입력하는 텍스트나 도는 시스템에 의해 나타나는 텍스트 중 하나입니다.

작업 제목을 표시하기 위한 기본적인 사용을 제외하고 중요한 새로운 용어를 처음 사용할 때 이텔릭체로 표시합니다. 예:

Publican은 *DocBook* 발행 시스템입니다.

1.2. 인용문 규정

터미널 출력 결과 및 소스 코드 목록은 주위의 문장에서 잘 보이는 위치에 설정됩니다.

터미널로 보내진 출력 결과는**mono-spaced roman**에 설정되어 다음과 같이 나타납니다:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

소스 코드 목록도 **mono-spaced roman**에 설정되지만, 다음과 같이 구문 강조가 추가되어 있습니다:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                 struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

1.3. 알림 및 경고

마지막으로, 3 종류의 시각적 스타일을 사용하여 간과될 수 있는 정보에 주의를 집중시킵니다.

참고

알림에서는 현재 작업에 대한 도움말, 지름길 또는 대안적 방법을 제공합니다. 알림 내용을 무시해도 상관없지만 효율적으로 작업할 수 있는 방법을 놓칠 수 있습니다.

중요

중요 상자에서는 현재 세션에만 적용되는 설정을 변경하거나 업데이트를 적용하기 전 다시 시작해야 하는 서비스와 같이 간과하기 쉬운 세부 사항을 제공합니다. 중요 상자를 무시해도 데이터를 손실하게 되지 않지만 문제를 일으킬 수 있습니다.

주의

경고는 무시해서는 안됩니다. 경고를 무시할 경우 대부분 데이터가 손실될 수 있습니다.

2. 피드백

오자를 발견하셨거나 보다 좋은 메뉴얼을 만들기 위한 제안이 있다면 언제든지 저희에게 알려주십시오. **Documentation-cluster**에 대한 리포트를 버그질라 (http://bugzilla.redhat.com/bugzilla/)에 제출해 주십시오.

문서 식별자를 반드시 기입해 주시기 바랍니다:

```
Virtual_Server_Administration(EN)-6 (2010-10-14T16:28)
```

문서 식별자를 기입하실 때 버전 번호를 정확히 기입해 주십시오.

문서 자료 개선을 위한 제안이 있으시면, 최대한 상세히 설명해 주시기 바랍니다. 오류를 발견하셨다면, 저희가 쉽게 식별할 수 있도록 장/절 번호와 오류 주변의 문장들을 함께 보내주시기 바랍니다.

1장. 로드 밸런서 추가 기능 개요

로드 밸런서 추가 기능은 실제 서버들 사이의 IP 로드밸런싱을 위해LVS (Linux Virtual Server)를 제공하는 통합된 소프트웨어 구성 요소 모음입니다. 로드 밸런서 추가 기능은 동일하게 설정된 한 쌍의 컴퓨터 상에서 실행됩니다: 둘 중 하나는 *활성 LVS 라우터*이고 다른 하나는 *백업 LVS 라우터*이어야 합니다. 활성 LVS 라우터는 두 가지 역할을 수행합니다:

- 실제 서버들 사이의 로드 밸런싱.
- 각각의 실제 서버에서 서비스 무결성 확인.

백업 LVS 라우터는 활성 LVS 라우터를 모니터하고 활성 LVS 라우터에 장애가 발생할 경우 이를 대신합니다.

다음 장에서는 로드 밸런서 추가 기능 구성 요소 및 기능에 대한 개요를 다루고 있으며 이는 다음과 같은 절로 구성되어 있습니다:

1.1. 기본 로드 밸런서 추가 기능 설정

그림 1.1. “기본 로드 밸런서 추가 기능 설정 ”는 두 개의 레이어로 되어있는 간단한 로드 밸런서 추가 기능 설정을 보여줍니다: 첫 번째 레이어는 두 개의 LVS 라우터 — 즉 하나의 활성 LVS 라우터와 하나의 백업 LVS 라우터로 구성되어 있습니다. 각각의 LVS 라우터에는 두 개의 네트워크 인터페이스가 있는데 하나는 인터넷에 연결되어 있고 다른 하나는 사설 네트워크에 연결되어 있어서 두 개의 네트워크 사이의 소통량을 조절할 수 있습니다. 예에서 활성 라우터는 *네트워크 주소 변환 (Network Address Translation)* 즉 *NAT*를 사용하여 인터넷으로 부터 두 번째 레이어에 있는 여러 실제 서버로 오가는 소통량을 제어하며, 이 서버들은 실제 서비스를 제공합니다. 따라서, 이 예에서 실제 서버는 전용 사설 네트워크 부분에 연결되어 모든 공개 트래픽을 활성 LVS 라우터를 통해 전해주고 받게 됩니다. 외부에서 볼 때 이 서버들은 하나의 엔티티처럼 보입니다.

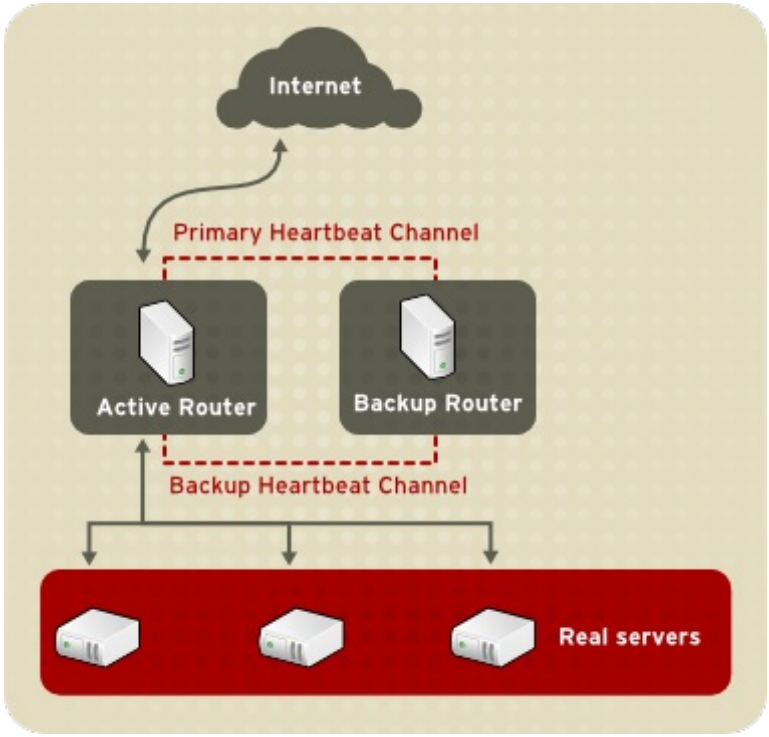

그림 1.1. 기본 로드 밸런서 추가 기능 설정

LVS 라우터로 들어온 서비스 요청은 *가상 IP* 주소, 즉 *VIP*로 주소가 부여됩니다. 이는 사이트의 관리자가 www.example.com와 같은 FQDN (fully-qualified domain name)을 부여한 공개적으로 라우팅할 수 있는 주소이며, 하나 이상의 *가상 서버*를 이 주소에 지정합니다. 가상 서버는 특정 가상 IP에서 사용할 수 있도록 설정된 서비스입니다. **Piranha Configuration Tool**를 사용하여 가상 서버를 설정하는 방법에 대한 자세한 내용은 4.6절. “VIRTUAL SERVER”에서 참조하기 바랍니다. VIP 주소는 장애 조치를 하는 동안 하나의 LVS 라우터에서 다른 라우터로 이전되기 때문에 해당 IP 주소의 존재는 계속 유지됩니다(이를 *유동 IP 주소*라고 합니다).

VIP 주소는 LVS 라우터를 인터넷으로 연결시키는 장치와 같은 장치로 알리아스(alias)될 수 있습니다. 예를 들어, eth0가 인터넷에 연결되어 있을 경우, 다중 가상 서버는 **eth0:1**로 알리아스될 수 있습니다. 다른 방법으로 각각의 가상 서버를 서비스마다 분리된 장치와 연관시킬 수 있습니다. 예를 들면, HTTP 트래픽은 **eth0:1**에서 처리될 수 있으며, FTP 트래픽은 **eth0:2**에서 처리될 수 있습니다.

한 번에 하나의 LVS 라우터만이 활성화됩니다. 활성 라우터의 역할은 가상 IP 주소에서 실제 서버로 서비스 요청을 경유시키는 것입니다. 이러한 작업은 1.3절. “로드 밸런서 추가 기능 스케줄링 개요 ”에 자세히 설명되어 있는 8가지의 지원 가능한 로드 밸런싱 알고리즘 중 하나에 기초하여 이루어 집니다.

또한, 활성 라우터는 *send/expect 스크립트*를 통해 실제 서버에 있는 특정 서비스의 전반적인 상태를 동적으로 모니터합니다. HTTPS나 SSL과 같은 동적인 데이터를 요구하는 서비스의 상태를 점검하기 위해 관리자는 외부 프로그램을 실행 할 수도 있습니다. 실제 서버에서의 서비스가 잘 작동하지 않을 경우, 활성 라우터는 서버가 정상적인 작동 상태로 돌아올 때 까지 실제 서버로의 작업 전송을 중단합니다.

백업 LVS 라우터는 대기 시스템의 역할을 실행합니다. 주기적으로 LVS 라우터는 주요 외부 공개 인터페이스를 통해서 하트비트 메시지 (heartbeat message)를 교환하며, 장애시에는 전용 인터페이스를 통해서도 하트비트 메시지를 교환합니다. 백업 LVS 라우터는 정해진 간격 내로 하트비트 메시지를 받지 못하면, 장애 조치를 시작하고 활성 LVS 라우터의 역할을 실행하려 합니다. 장애 조치 작업이 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.

그림 1.1. “기본 로드 밸런서 추가 기능 설정 ”에 있는 단순한 2중 레이어 설정은 자주 변하지 않는 데이터를 제공하는 경우에 아주 적합합니다. — 정적인 웹 페이지가 한 예입니다 — 이는 각각의 실제 서버가 노드사이에 자동으로 데이터를 동기화하지 않기 때문입니다.

1.1.1. 실제 서버에서 데이터 복제 및 데이터 공유

실제 서버 간의 동일한 데이터를 공유하기 위해 로드 밸런서 추가 기능에 내장된 요소가 없기 때문에, 관리자가 할 수 있는 두가지 기본적인 옵션이 있습니다:

- 실제 서버 풀을 통한 데이터 동기화
- 공유 데이터 액세스에 필요한 토폴로지에 세 번째 레이어 추가

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

1.1.1.1. 데이터 동기화를 위한 실제 서버 설정

관리자가 실제 서버 풀에서 데이터를 동기화하는 방법에는 여러 가지가 있습니다. 예를 들어, 쉘 스크립트를 사용하여 동시에 실제 서버에 업데이트된 웹 페이지를 게시할 수 있습니다. 또한, 시스템 관리자는 **rsync**와 같은 프로그램을 사용하여 일정 간격으로 모든 노드에 걸쳐 변경된 데이터를 복사할 수 있습니다.

하지만, 사용자가 수시로 파일을 업로드하거나 데이터베이스를 처리해야 해서, 작업 설정이 과부하되는 상태에서 이런 데이터 동기화 유형을 사용하는 것은 최적으로 동작하지 못합니다. 따라서, 과도한 작업량이 있는 설정에서는 *3-tier 토폴로지*를 사용하는 것이 좋습니다.

1.2. 3-Tier 로드 밸런서 추가 기능 설정

그림 1.2. “3-Tier 로드 밸런서 추가 기능 설정 ”는 전형적인 3-tier 로드 밸런서 추가 기능 토폴로지를 보여주고 있습니다. 예에서, 활성 LVS 라우터는 인터넷에서 들어오는 요청을 실제 서버 풀로 라우트합니다. 그 후 각각의 실제 서버는 네트워크를 통해 공유 데이터 소스에 액세스합니다.

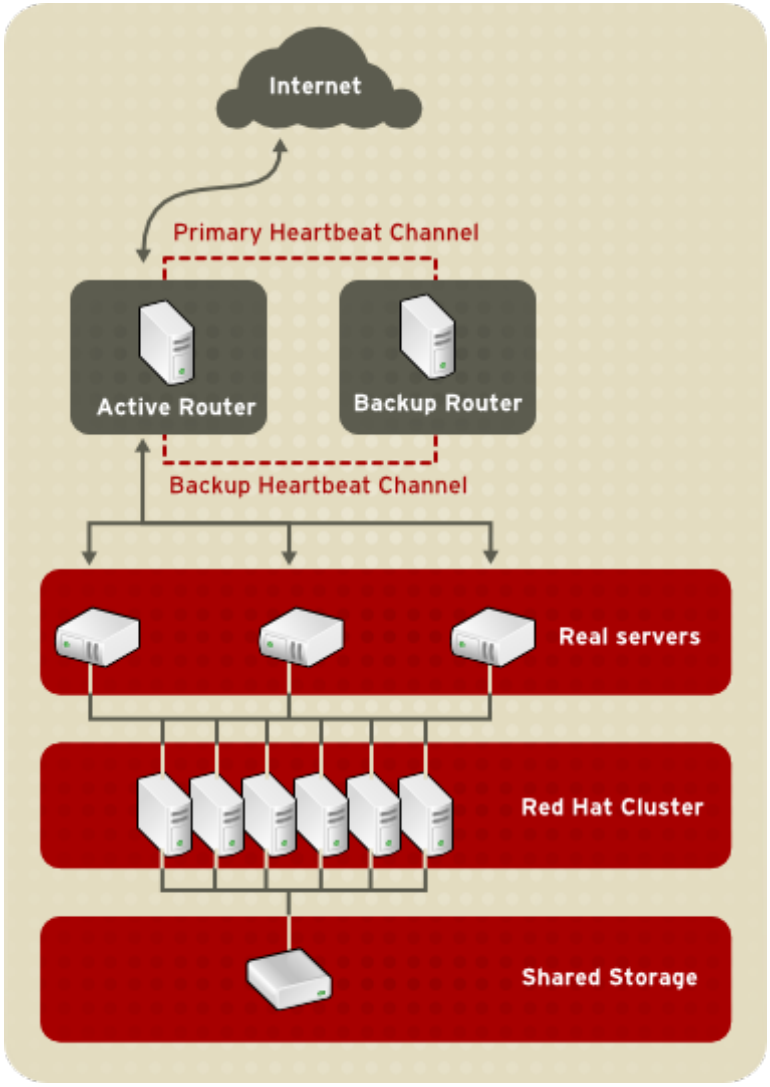

그림 1.2. 3-Tier 로드 밸런서 추가 기능 설정

이 설정은 데이터 송수신 량이 많은 FTP 서버에 아주 적합합니다. 여기서 액세스할 수 있는 데이터는 중앙 고가용성 서버에 저장되어 있으며, 각각의 실제 서버들은 외부로 내보내진 NFS 디렉토리나 Samba 공유를 통해 데이터를 억세스합니다. 이러한 토폴로지는 또한 트랜잭션을 위한 중앙 고가용성 데이터베이스를 액세스하는 웹사이트에도 권장합니다. 또한, Red Hat Cluster Manager를 사용한 active-active 설정으로, 관리자는 하나의 고가용성 클러스터를 설정하여 동시에 두 가지 역할을 담당하게 할 수 있습니다.

예에서 세 번째 레이어에서는 Red Hat Cluster Manager를 사용할 필요가 없지만 고가용성 솔루션 사용에 문제가 발생하면 단일 장애 지점(single point of failure,SPF)를 초래할 수 있습니다.

1.3. 로드 밸런서 추가 기능 스케줄링 개요

로드 밸런서 추가 기능 사용 장점 중 하나는 실제 서버 풀에서 유연성있게 IP 수준 로드 밸런싱을 실행할 수 있다는 것입니다. 이는 로드 밸런서 추가 기능 설정 시 관리자가 스케줄링 알고리즘을 유연성있게 선택할 수 있기 때문입니다. 로드 밸런서 추가 기능 로드 밸런싱은 DNS 시스템의 계층적 특성과 클라이언트의 캐쉬 때문에 부하가 불균등하게 분산되는 *라운드 로빈 (Round-Robin) DNS*와 같은 유연하지 못한 방식 보다 우월합니다. 또한 LVS 라우터에 의해 사용되는 저수준 (low-level) 필터링 기술은 어플리케이션 수준 요청 포워딩보다 이점을 가집니다. 이는 네트워크 패킷 수준에서의 로드 밸런싱이 계산 오버헤드를 최소화하고 더 나은 확장성을 제공하기 때문입니다.

스케줄링을 사용하여 활성 라우터는 실제 서버의 작동 내용과 옵션으로 라우팅 서비스 요청시 관리자 지정

가중치 요인을 고려할 수 있습니다. 지정된 가중치를 사용하여 개별적 컴퓨터에 임의로 우선 순위를 부여할 수 있습니다. 스케줄링 형식을 사용하면, 다양한 하드웨어 및 소프트웨어를 결합하여 실제 서버 그룹을 생성할 수 있으며 활성 라우터는 각각의 실제 서버를 균등하게 로드할 수 있게 됩니다.

로드 밸런서 추가 기능의 스케줄링 메카니즘은 *IP 가상 서버* 또는 *IPVS* 모듈이라는 커널 패치에 의해 제공됩니다. 이러한 모듈은 단일 IP 주소에 있는 다중 서버와 잘 작동하게 고안된 *레이어 4* (*L4*) 전송 레이어 스위칭을 활성화합니다.

실제 서버로 패킷을 효과적으로 추적하고 라우트하기 위해 IPVS는 커널에 *IPVS 테이블*을 만듭니다. 이러한 테이블은 가상 서버에서의 요청을 되돌리고 풀에 있는 실제 서버에서 복귀하기 위해 활성 LVS 라우터에 의해 사용됩니다. IPVS 테이블은 *ipvsadm* — 사용 가능성에 따라 클러스터 멤버를 추가 또는 삭제 유틸리티에 의해 지속적으로 업데이트됩니다.

1.3.1. 스케줄링 알고리즘

주어진 가상 서버에 대해 관리자가 선택하는 스케줄링 알고리즘에 따라 IPVS 테이블 구조가 결정됩니다. 클러스터할 서비스 유형 및 서비스 스케줄 방식에서 유연성을 극대화하기 위해 Red Hat Enterprise Linux는 다음과 같은 스케줄링 알고리즘을 제공합니다. 스케줄링 알고리즘을 지정하는 방법에 대한 자세한 내용은 4.6.1절. “**VIRTUAL SERVER** 하위 섹션”에서 참조하시기 바랍니다.

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. 이러한 스케줄링 모델은 라운드 로빈 DNS 방식과 유사하지만 호스트 기반이 아닌 네트워크 연결 기반의 보다 세분화된 방식입니다. 또한 로드 밸런서 추가 기능 라운드 로빈 스케줄링은 캐쉬된 DNS 쿼리에 의해 불균등하게 분산되지 않습니다.

가중치 부여된 라운드 로빈 (Weighted Round-Robin) 스케줄링

각각의 요청을 실제 서버의 풀로 순차적으로 배분하는 방식이지만, 처리 용량이 큰 서버에 더 많은 작업을 배분합니다. 처리 용량은 사용자 지정 가중치 요인에 의해 표시되며, 동적 부하 정보에 따라 상하로 조정됩니다. 실제 서버 가중치에 대한 보다 자세한 내용은 1.3.2절. “서버 가중치 및 스케줄링 ”에서 참조하시기 바랍니다.

가중치 기반 라우드 로빈 스케줄링은 풀에 있는 실제 서버의 용량이 다를 경우 선택할 수 있는 사항입니다. 하지만, 요청 부하량이 아주 다양할 경우, 더 가중치가 있는 서버가 요청 공유량보다 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, IPVS 테이블을 통해 실제 서버로의 라이브 연결을 추적하기 때문에 요청 부하에 극도로 다양한 처리량이 있을 경우 유용합니다. 각각의 노드에 대략 비슷한 처리 용량이 있는 실제 서버 풀의 경우에도 적합합니다. 서버 그룹에 다른 처리 용량이 있을 경우, 가중치 기반 최소 접속 스케줄링이 더 적합합니다.

가중치 부여된 최소 접속 (Weighted Least-Connections) 스케줄링 (기본값)

처리 용량 대비 활성화된 연결이 적은 서버로 더 많은 요청을 배분하는 방식입니다. 처리 용량은 사용자 지정 가중치에 의해 정해지며, 동적 부하 정보에 의해 상하로 조정됩니다. 실제 서버 풀에 다양한 처리 용량의 하드웨어가 있을 경우, 가중치를 포함하는 이러한 알로리즘이 가장 적합합니다. 가중치 기반 실제 서버에 관한 보다 자세한 내용은 1.3.2절. “서버 가중치 및 스케줄링 ”에서 참조하시기 바랍니다.

지역성 기반 최소 연결(Locality-Based Least-Connection,LBLC) 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 서버가 반부하 상태에 있지 않을 경우 IP 주소를 최소로 부하된 실제 서버로 할당하여 IP 주소의 패킷을 해당 주소의 서버로 라우팅합니다.

복제를 허용하는 지역성 기반 최소 연결(Locality-Based Least-Connection with Replication, LBLCR) 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 묶는 다는 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.

목적지 해쉬 (Destination Hash) 스케줄링

정적 해쉬 테이블에 있는 목저지 IP를 찾아 실제 서버의 풀로 요청을 배분하는 방식입니다. 이러한 알고리즘은 프록시-캐쉬 서버 클러스터에서의 사용을 위해 고안되었습니다.

소스 해쉬 (Source Hash) 스케줄링

정적 해쉬 테이블에 있는 소스 IP를 찾아 실제 서버의 풀로 요청을 배분하는 방식입니다. 이러한 알고리즘은 여러 방화벽이 있는 LVS 라우터에 사용됩니다.

1.3.2. 서버 가중치 및 스케줄링

로드 밸런서 추가 기능 관리자는 실제 서버 풀에 있는 각각의 노드에 *가중치*를 부여할 수 있습니다. 이러한 가중치는 정수 값으로 모든 *가중치 관련* 스케줄링 알고리즘 (예: 가중치 기반 최소 접속)에 영향을 미치며 LVS 라우터가 다른 처리 용량이 있는 하드웨어를 보다 균등하게 로드합니다.

가중치는 서로에 관해 비율로 동작합니다. 예를 들어, 하나의 실제 서버가 가중치 1로 되어 있고 다른 서버가 가중치 5로 되어 있을 경우, 가중치 1로 된 서버가 1번 연결될 때 마다 가중치 5로 되어 있는 서버는 5번 연결됩니다. 실제 서버 가중치에 해당하는 기본 값은 1입니다.

실제 서버 풀에 있는 다양한 하드웨어 설정 사항에 가중치를 추가하는 것이 보다 효과적으로 클러스터의 로드 밸런스를 유지할 수 있을 지라도 실제 서버가 실제 서버 풀에 소개될 때와 가중치 기반 최소 접속을 사용하여 가상 서버가 스케줄링될 때 일시적 부하 불균등 상태가 될 수 있습니다. 예를 들어, 실제 서버 풀에 세 개의 서버가 있다고 가정합시다. 서버 A와 B는 가중치 1이 부여되어 있고 세번째 서버 C는 가중치 2가 부여되어 있습니다. 어떤 이유로 서버 C가 다운되었을 경우 서버 A와 B는 균등하게 부하량을 배분하게 됩니다. 하지만, 서버 C가 온라인으로 돌아오면 LVS 라우터는 이 서버에 0 연결이 있다고 보고 서버 A 및 B와 동등한 수준이 될 때 까지 서버를 모든 들어오는 요청으로 채웁니다.

이러한 현상이 일어나지 않게 하기 위해, 관라자는 가상 서버를 *Quiesce* 서버로 만들 수 있습니다 — 새로운 실제 서버 노드가 온라인 상태로 될 때 마다 최소 접속 테이블이 0으로 재설정되어 LVS 라우터는 모든 실제 서버가 클러스터에 새로 추가된 것처럼 요청을 라우트합니다.

1.4. 라우팅 방식

Red Hat Enterprise Linux는 로드 밸런서 추가 기능 용 *NAT (Network Address Translation)* 또는 *NAT 라우팅*을 사용하여 관리자로 하여금 하드웨어를 사용하여 기존 네트워크로 로드 밸런서 추가 기능을 통합할 때 유연성을 제공합니다.

1.4.1. NAT 라우팅

그림 1.3. “NAT 라우팅을 사용하는 로드 밸런서 추가 기능 ”에서는 인터넷과 사설 네트워크 사이에서의 요청을 이동시키기 위해 NAT 라우팅을 사용한 로드 밸런서 추가 기능을 보여주고 있습니다.

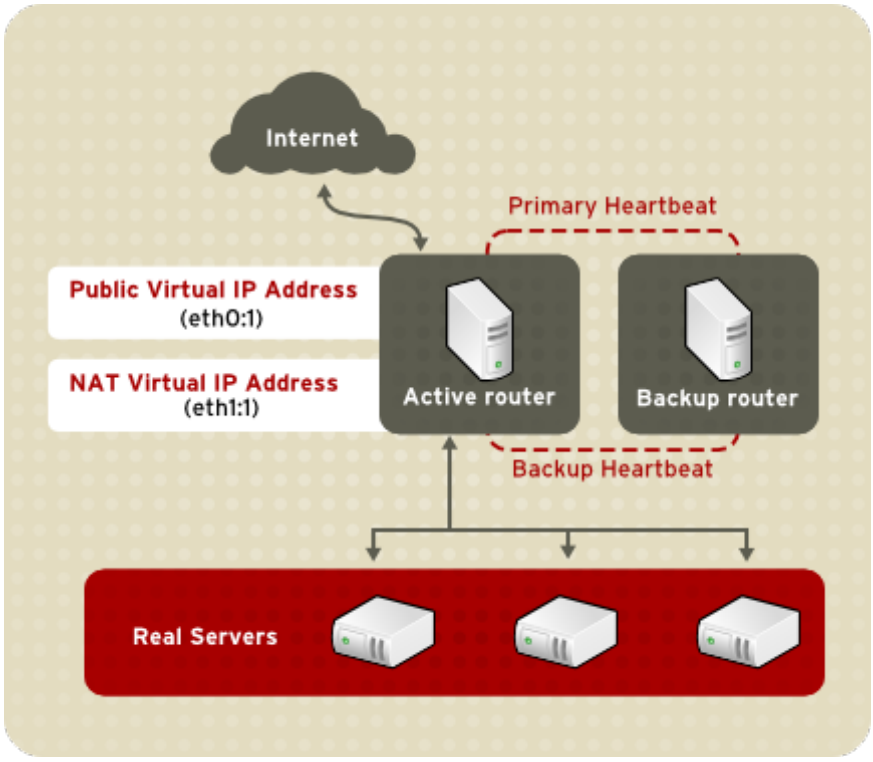

그림 1.3. NAT 라우팅을 사용하는 로드 밸런서 추가 기능

예에서 활성화된 LVS 라우터에는 두개의 NIC가 있습니다. 인터넷용 NIC는 eth0에 *실제 IP 주소*가 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.

예에서, LVS 라우터의 공개 LVS 유동 IP 주소 및 개인 NAT 유동 IP 주소는 두 개의 물리적 NIC로 알리아스됩니다. 각각의 유동 IP 주소를 LVS 라우터 노드에 있는 물리적 장치에 연결시킬 수 있지만, 필수적으로 두 개 이상의 NIC가 있어야 하는 것은 아닙니다.

이러한 토플로지를 사용하여 활성 LVS 라우터는 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 마스커레이딩*이라고 부릅니다.

NAT 라우팅을 사용하면 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.

1.4.2. 직접 라우팅

직접 라우팅을 사용하는 로드 밸런서 추가 기능 설정은 다른 로드 밸런서 추가 기능 네트워킹 토폴로지와 비교하여 성능이 향상되었습니다. 직접 라우팅으로 실제 서버는 LVS 라우터를 통해 외부로 나가는 패킷을 전달하지 않고 요청 중인 사용자에게 직접 패킷을 처리하고 보냅니다. 직접 라우팅은 LVS 라우터의 작업을 들

어오는 패킷이 처리하게 하여 네트워크 성능 문제 발생을 감소시켰습니다.

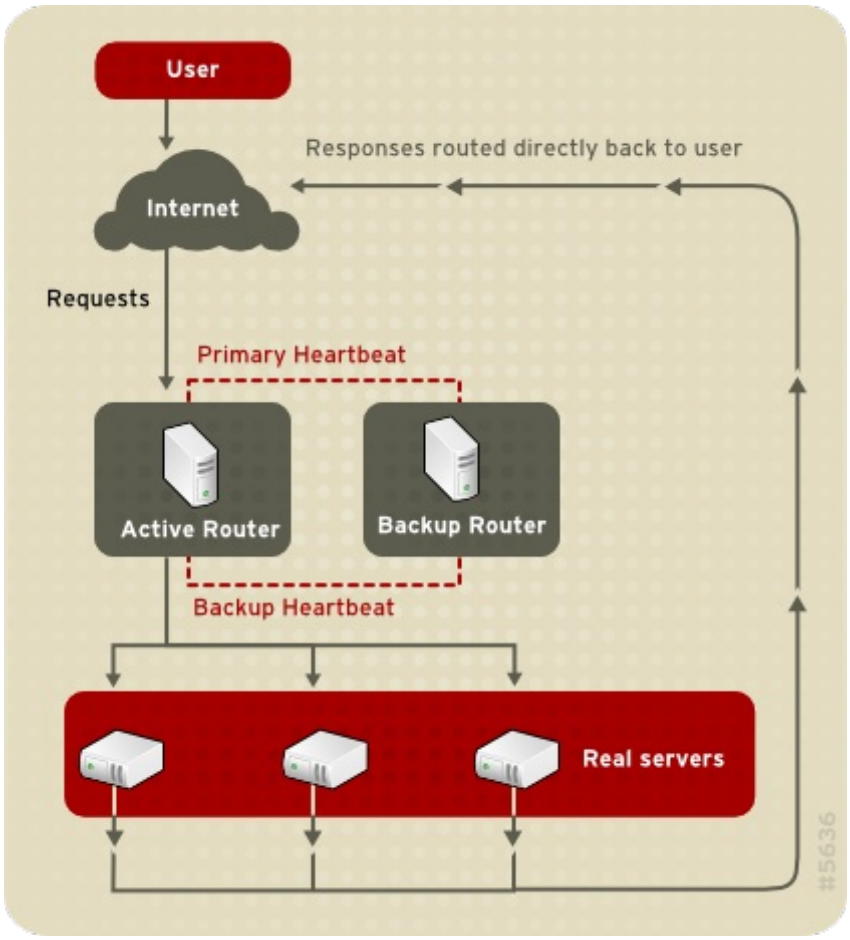

그림 1.4. 직접 라우팅을 사용하는 로드 밸런서 추가 기능

전형적인 직접 라우팅 로드 밸런서 추가 기능 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 보내기위해 실제 서버에서 확장성을 감안한 직접 라우팅이 추가될 수 있어 과중한 네트워크 로드에서 병목 현상이 나타날 수 있습니다.

1.4.2.1. 직접 라우팅 및 ARP 한계

로드 밸런서 추가 기능에서 직접 라우팅을 사용하는데 있어서 여러 장점이 있는 반면 제한점도 있습니다. 직접 라우팅응 통한 로드 밸런서 추가 기능 사용에서 가장 일반적으로 발생하는 문제는 *ARP* (Address Resolution Protocol)와의 문제입니다.

일반적으로 인터넷 상의 클라이언트는 IP 주소로 요청을 보냅니다. 네트워크 라우터는 IP 주소와 ARP가 있는 컴퓨터의 MAC 주소에 따라 수신지로 요청을 보냅니다. ARP 요청은 네트워크 상에 연결된 모든 컴퓨터로 브로드캐스트되고 올바른 IP/MAC 주소로된 컴퓨터는 패킷을 받습니다. IP/MAC은 ARP 캐시에 저장되어 주기적으로 (주로 15분 마다) 삭제되며 IP/MAC 주소로 다시 채워집니다.

직접 라우팅 로드 밸런서 추가 기능 설정에 있는 ARP 요청에서의 문제는 IP 주소로의 클라이언트 요청이 처리될 요청에 대한 MAC 주소와 관련되어 있어야 하며, 로드 밸런서 추가 기능 시스템의 가상 IP 주소도 MAC 주소와 관련되어 있어야 한다는 점입니다. 하지만, LVS 라우터와 실제 서버의 VIP가 같을 경우, ARP 요청은 VIP와 관련되어 있는 모든 컴퓨터로 브로드캐스트됩니다. 이는 VIP가 실제 서버 중 하나에 관련되어 직접적으로 요청을 처리하는 것, LVS 라우터를 바이패싱하여 로드 밸런서 추가 기능 설정 목적을 의미없게 하는 것과 같이 여러가지 문제를 발생시킬 수 있습니다.

이러한 문제를 해결하기 위해 들어오는 요청은 항상 실제 서버가 아닌 LVS 라우터로 보내야 합니다. 이러한 작업은 다음과 같은 이유로 **arptables_jf** 또는 **arptables** 패킷 필터링 도구를 사용하여 실행합니다:

- **arptables_jf**는 ARP가 VIP를 실제 서버와 관련짓지 않게 합니다.
- **iptables** 방식은 첫번째의 실제 서버에 있는 VIP 설정을 하지 않음으로 ARP 문제를 전적으로 회피합니다.

직접 라우팅 로드 밸런서 추가 기능 환경에서 **arptables** 또는 **iptables**을 사용하는 것에 대한 보다 자세한 내용은 3.2.1절. "직접 라우팅 및 **arptables_jf**" 또는 3.2.2절. "직접 라우팅 및 **iptables**"에서 참조하시기 바랍니다.

1.5. 지속성 및 방화벽 표시 기능

특정한 경우에는 로드 밸런서 추가 기능 (Load Balancer Add-On) 로드 밸런스 알고리즘이 가장 적합한 서버에 요청을 보내는것 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, 작업을 유지하기 위해 트랜잭션이 같은 서버에 의해 처리되지 않는 한 클라이언트가 제대로 작업할 수 없게 됩니다. 로드 밸런서 추가 기능은 *지속성 (persistence)* 및 *방화벽 표시 기능 (firewall marks)*이라는 두 가지 다른 종류의 기능을 사용하여 이를 처리합니다.

1.5.1. 지속성

지속성이 활성화되면 이는 타이머와 같은 기능을 수행합니다. 클라이언트가 서비스에 연결할 경우, 로드 밸런서 추가 기능은 특정 기간에 대한 마지막 연결을 기억하게 됩니다. 동일한 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 관리자가 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 포트로 되어있는 그룹 연결 문제를 처리하는데 지속성 기능이 가장 효과적인 방법은 아닙니다. 이러한 경우, *방화벽 표시 기능 (firewall marks)*을 사용하는 것이 가장 좋습니다.

1.5.2. 방화벽 표시 기능

방화벽 표시 기능은 프로토콜이나 연관된 프로토콜 그룹에 사용되는 가장 쉽고 효과적인 그룹 포트 방법입니다. 예를 들어, 로드 밸런서 추가 기능이 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 대한 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 로드 밸런서 추가 기능 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.

1.6. 로드 밸런서 추가 기능 — 블록 다이어그램

LVS 라우터는 프로그램 모음을 사용하여 클러스터 멤버 및 클러스터 서비스를 모니터합니다. 그림 1.5. "로드 밸런서 추가 기능 구성 요소"에서는 이러한 다양한 프로그램이 활성 및 백업 LVS 라우터에서 클러스터를 관리하기 위해 어떻게 함께 작동하는 지를 보여주고 있습니다.

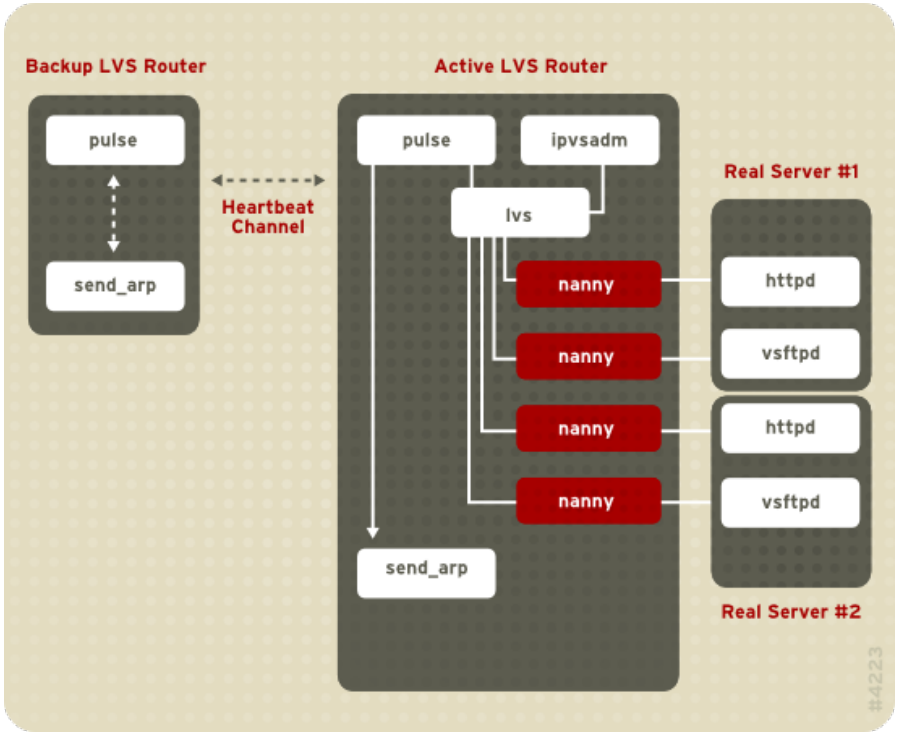

그림 1.5. 로드 밸런서 추가 기능 구성 요소

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

1.6.1. 로드 밸런서 추가 기능 구성 요소

1.6.1.1절. “**pulse**”는 LVS 라우터에 있는 각각의 소프트웨어 구성 요소에 대한 자세한 목록을 보여줍니다.

1.6.1.1. pulse

이는 LVS 라우터와 관련된 기타 모든 데몬을 시작하는 제어 프로세스입니다. 부팅시 **/etc/rc.d/init.d/pulse** 스크립트에 의해 데몬이 시작되면 **/etc/sysconfig/ha/lvs.cf** 설정 파일을 읽습니다. 활성 라우터에서 **pulse**는 LVS 데몬을 시작하며 백업 라우터에서 **pulse**는 사용자 설정 가능한 간격으로 간단한 하트비트(heartbeat)를 실행하여 활성 라우터의 상태를 확인합니다. 사용자 설정 가능 간격 후에 활성 라우터가 응답 실패할 경우, 장애 조치를 시작합니다. 장애 조치 중 백업 라우터에 있는 **pulse**는 활성 라우터에 있는 **pulse** 데몬을 지시하여 모든 LVS 서비스를 종료하고 **send_arp** 프로그램을 시작하여 백업 라우터의 MAC 주소로 유동 IP 주소를 다시 지정하고 **lvs** 데몬을 시작합니다.

1.6.1.2. lvs

lvs 데몬이 일단 **pulse**에 의해 호출되면 활성화된 LVS 라우터에서 실행됩니다. 이는 **/etc/sysconfig/ha/lvs.cf** 설정 파일을 읽으며, **ipvsadm** 유틸리티를 호출하여 IPVS 라우팅 테이블을 생성하고 관리하며 각각의 설정된 로드 밸런서 추가 기능 서비스 용 **nanny** 프로세스를 지정합니다. **nanny**가 실제 서버가 다운되었다고 보고하면 **lvs**가 **ipvsadm** 유틸리티를 지시하여 IPVS 라우팅 테이블에서 실제 서버를 삭제합니다.

1.6.1.3. ipvsadm

이 서비스로 커널에서 IPVS 라우팅 테이블을 업데이트합니다. **lvs** 데몬은 IPVS 라운팅 테이블에 있는 항목을 추가, 변경, 삭제하기 위해 **ipvsadm** 명령을 호출하여 로드 밸런서 추가 기능을 설정 및 관리합니다.

1.6.1.4. nanny

nanny 모니터링 데몬은 활성 LVS 라우트에서 실행됩니다. 이 데몬을 통해, 활성화된 라우터는 실제 서버 상태를 확인하며, 옵션으로 작업량을 모니터합니다. 분리된 프로세스는 실제 서버에 정의된 각각의 서비스에 대해 실행됩니다.

1.6.1.5. /etc/sysconfig/ha/lvs.cf

이는 로드 밸런서 추가 기능 설정 파일입니다. 직접 또는 간접적으로 모든 데몬은 이 파일에서 설정 정보를 얻게 됩니다.

1.6.1.6. Piranha Configuration Tool

로드 밸런서 추가 기능 모니터링, 설정, 관리 용 웹 기반 도구로 **/etc/sysconfig/ha/lvs.cf** 로드 밸런서 추가 기능 설정 파일을 관리하기 위한 기본값 도구입니다.

1.6.1.7. send_arp

이 프로그램은 장애조치 중 하나의 노드에서 다른 노드로 유동 IP 주소가 변경되었을때 ARP 브로드캐스트를 전송합니다.

2장. 로드 밸런서 추가 기능 (Load Balancer Add-On) 초기 설정 는 Red Hat Enterprise Linux를 LVS 라우터로 설정하기 전에 따라야만 하는 설치 후 설정 단계를 재확인합니다.

2장. 로드 밸런서 추가 기능 (Load Balancer Add-On) 초기 설정

Red Hat Enterprise Linux를 설치한 후에, LVS 라우터 및 실제 서버를 설정하기 위해 기본적인 설정 단계를 따르셔야 합니다. 다음 부분에서는 초기 설정 단계에 대해 자세히 설명합니다.

알림

로드 밸런서 추가 기능을 시작하여 활성화 노드가된 LVS 라우터 노드를 *1차* 노드라고 합니다. 로드 밸런서 추가 기능을 설정할 때 1차 노드에서 **Piranha Configuration Tool**를 사용합니다.

2.1. LVS 라우터 상의 설정 서비스

Red Hat Enterprise Linux 설치 프로그램으로 로드 밸런서 추가 기능 설정에 필요한 모든 요소를 설치할 수 있지만, 로드 밸런서 추가 기능을 설정하기 전에 필요한 서비스를 활성화시켜야 합니다. 두 LVS 라우터에 대해 부팅시 시작할 서비스를 설정합니다. Red Hat Enterprise Linux에서 부팅시 활성화할 세 개의 서비스 설정 도구가 있습니다: 명령행 프로그램 **chkconfig**, ncurses 기반 프로그램 **ntsysv**, 그래픽 **Services Configuration Tool**. 이러한 도구를 사용하시려면 root 액세스가 필요합니다.

알림

root 액세스를 얻으려면 쉘 프롬프트를 열고 **su -** 명령 뒤에 root 암호를 입력합니다. 예:

```
$ su - root password
```

LVS 라우터에서 부팅시 활성화되도록 설정해야 할 세 가지의 서비스가 있습니다:

- **piranha-gui** 서비스 (1차 노드에서만)
- **pulse** 서비스
- **sshd** 서비스

다중 포트 서비스를 클러스터링하거나 또는 방화벽 표시 기능을 사용할 경우, **iptables** 서비스를 활성화해야 합니다.

이러한 서비스가 런레벨 3과 5에서 활성화되도록 설정하는 것이 좋습니다. 이를 위해 **chkconfig**를 사용하며 각각의 서비스에 대해 다음과 같은 명령을 입력합니다:

/sbin/chkconfig --level 35 *데몬* on

위의 명령에서, ***daemon***을 활성화하려는 서비스의 이름으로 대체합니다. 시스템 상의 서비스 목록을 얻고, 어떤 런레벨에서 활성화되도록 설정되었는지를 알려면 다음과 같은 명령을 실행합니다:

/sbin/chkconfig --list

경고

chkconfig를 사용하여 위의 서비스를 활성화하는 것으로 데몬이 시작되지 않습니다. 이를 위해 **/sbin/service** 명령을 사용합니다. 2.3절. "**Piranha Configuration Tool** 서비스 시작 "에서 **/sbin/service** 명령을 사용하는 방법을 참조하시기 바랍니다.

런레벨 및 **ntsysv**와 **Services Configuration Tool**를 사용한 서비스 설정에 관한 자세한 내용은 *Red Hat Enterprise Linux System Administration Guide*에 있는 *"서비스 액세스 제어"* 장을 참조하시기 바랍니다.

2.2. Piranha Configuration Tool 암호 설정

주요 LVS 라우터에서 처음으로 **Piranha Configuration Tool**를 사용하기 전에 암호를 설정하여 이 라우터로의 액세스를 제한해야 합니다. 이를 실행하기 위해 root로 로그인하여 다음과 같은 명령을 사용합니다:

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

이러한 명령을 입력한 후에 프롬프트에서 관리자 암호를 설정합니다.

경고

암호의 보안을 강화하려면 고유명사, 자주 사용되는 약어, 사전에 있는 단어 등을 암호에 포함시키지 않습니다. 시스템 상에 암호가 풀린 상태로 두지 않습니다.

Piranha Configuration Tool를 활성화하는 도중 암호가 변경되었을 경우, 관리자는 새로운 암호를 알려주어야 합니다.

2.3. Piranha Configuration Tool 서비스 시작

Piranha Configuration Tool에 대한 암호를 설정한 후에 **/etc/rc.d/init.d/piranha-gui**에 있는 **piranha-gui** 서비스를 시작 또는 재시작합니다. 이를 실행하기 위해 root로 다음과 같은 명령을 입력합니다:

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

또는

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

이 명령을 실행하면 **/usr/sbin/piranha_gui -> /usr/sbin/httpd** 심볼릭 링크를 호출하여 Apache HTTP Server의 개인적 세션을 시작하게 됩니다. 보안상의 이유로 **httpd**의 **piranha-gui** 버전은 분리된 piranha 사용자 프로세스로 실행됩니다. **piranha-gui**가 **httpd** 서비스를 활용한다는 것은 다음을 의미합니다:

1. 시스템에 Apache HTTP Server가 설치되어 있어야 합니다.
2. **service** 명령으로 Apache HTTP Server를 정지하거나 재시작하면 **piranha-gui** 서비스가 중지됩니다.

경고

/sbin/service httpd stop 또는 **/sbin/service httpd restart** 명령이 LVS 라우터에서 실행될 경우, 다음과 같은 명령을 사용하여 **piranha-gui** 서비스를 시작해야 합니다. **/sbin/service piranha-gui start**

piranha-gui 서비스는 로드 밸런서 추가 기능을 설정하기 위해 필요합니다. 하지만, 원격으로 로드 밸런서 추가 기능을 설정할 경우, **sshd** 서비스도 필요합니다. **Piranha Configuration Tool**를 사용한 설정이 완료될 때 까지 **pulse** 서비스를 시작할 필요가 *없습니다.* **pulse** 서비스를 시작하는 방법에 관한 내용은 4.8절. “로드 밸런서 추가 기능 시작 ”에서 참조하시기 바랍니다.

2.3.1. Piranha Configuration Tool 웹 서버 포트 설정

Piranha Configuration Tool는 기본값으로 포트 3636에서 실행됩니다. 이 포트 번호를 변경하려면 **piranha-gui** 웹 서버 설정 파일 **/etc/sysconfig/ha/conf/httpd.conf**의 섹션 2에 있는 **Listen 3636** 행을 변경합니다.

Piranha Configuration Tool를 사용하려면 최소 텍스트 전용 웹 브라우저가 있어야 합니다. 주요 LVS 라우터에서 웹 브라우저를 시작할 경우 **http://*localhost*:3636**을 엽니다. ***localhost***를 주요 LVS 라우터의 호스트명이나 IP 주소로 대체하여 웹 브라우저에서 **Piranha Configuration Tool**를 사용할 수 있습니다.

브라우저가 **Piranha Configuration Tool**에 연결되면 설정 서비스를 액세스하기 위해 로그인해야 합니다. **Username** 란에 **piranha**를 입력하고 **piranha** 란에 **piranha-passwd**를 사용하여 암호를 설정합니다.

Piranha Configuration Tool가 실행되면 네트워크를 통해 이 도구로의 액세스를 제한할 수 있습니다. 다음 부분에서는 이를 실행하는 방법에 대해 설명합니다.

2.4. Piranha Configuration Tool로의 액세스 제한

Piranha Configuration Tool는 사용 가능한 사용자명과 암호를 요구합니다. **Piranha Configuration Tool**로 전달된 모든 데이터가 평문으로 되어 있으므로 신뢰하는 네트워크나 또는 로컬 컴퓨터로의 액세스만으로 제한할 것을 권장합니다.

액세스를 제한하는 가장 쉬운 방법은 **/etc/sysconfig/ha/web/secure/.htaccess** 파일을 편집하여 Apache HTTP Server의 내장된 액세스 제어 메카니즘을 사용하는 것입니다. 파일을 변경한 후에 서버가 디렉토리로 액세스할 때 마다 **.htaccess** 파일을 확인하기 때문에 **piranha-gui** 서비스를 재시작하지 않아도 됩니다.

디렉토리로의 액세스 제한 기본값은 누구나 디렉토리에 있는 내용물을 볼 수 있게 되어 있습니다. 기본값 액세스는 다음과 같이 나타납니다:

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

로컬 호스트에게만 **Piranha Configuration Tool**의 액세스를 제한하기 위해 **.htaccess** 파일을 변경하여 루프백 장치 (127.0.0.1)에서만 액세스되게 합니다. 루프백 장치에 관한 보다 자세한 내용은 *Red Hat Enterprise Linux Reference Guide*에 있는 *네트워크 스크립트* 장에서 참조하시기 바랍니다.

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

예에서 볼 수 있듯이 특정 호스트나 서브넷을 허용할 수 있습니다:

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

예에서 192.168.1.100의 IP 주소를 갖는 컴퓨터와 172.16.57/24 네트워크 상의 컴퓨터의 웹 브라우져만이 **Piranha Configuration Tool**에 액세스할 수 있습니다.

경고

Piranha Configuration Tool **.htaccess** 파일을 편집하면 **/etc/sysconfig/ha/web/secure/** 디렉토리에 있는 설정 페이지로의 액세스가 제한될 수 있으나 로그인과 **/etc/sysconfig/ha/web/**에 있는 도움말 페이지로의 액세스가 제한되지는 않습니다. 이러한 디렉토리로의 액세스를 제한하기 위해 **/etc/sysconfig/ha/web/secure/.htaccess** 파일과 동일하게 **order**, **allow**, **deny**를 사용하여 **/etc/sysconfig/ha/web/** 디렉토리에 있는 **.htaccess** 파일을 생성합니다.

2.5. 패킷 포워딩 활성화

LVS 라우터가 실제 서버에 네트워크 패킷을 올바르게 포워딩하기 위해 각각의 LVS 라우터 노드는 커널에서 IP 포워딩을 활성화 시켜야 합니다. root로 로그인하여 **/etc/sysctl.conf**에 있는 **net.ipv4.ip_forward = 0** 행을 다음과 같이 변경합니다:

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

시스템 재부팅 시 변경 사항을 적용합니다.

IP 포워딩이 활성화 되었는지를 확인하기 위해 root로 다음과 같은 명령을 실행합니다:

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

위의 명령으로 **1**이 출력되면 IP 포워딩이 활성화된 것입니다. **0** 값이 출력될 경우, 다음과 같은 명령을 사용하여 수동으로 IP 포워딩을 활성화시킬 수 있습니다:

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

2.6. 실제 서버에서 서비스 설정

실제 서버가 Red Hat Enterprise Linux 시스템일 경우, 부팅 시 활성화할 적절한 서버 데몬을 설정합니다. 이러한 데몬에는 웹 서비스 용 **httpd**나 FTP 또는 Telnet 서비스 용 **xinetd**가 있습니다.

원격으로 실제 서버를 액세스하여 **sshd** 데몬을 설치하고 실행하는 것이 유용할 수 도 있습니다.

3장. 로드 밸런서 추가 기능 설정

로드 밸런서 추가 기능은 LVS 라우터와 실제 서버라는 두 개의 기본 그룹으로 구성되어 있습니다. 단일 오류 지점(single point of failure,SPF)을 방지하려면, 각각의 그룹에는 최소 두 개의 멤버 시스템이 있어야 합니다.

LVS 라우터 그룹은 Red Hat Enterprise Linux에서 실행되고 있는 두 개의 동일한 또는 유사한 시스템으로 구성되어 있어야 합니다. 하나의 시스템이 긴급 대기 모드에 있을 동안 다른 하나는 활성 LVS 라우터로 실행되어 두 개의 시스템이 가능한 동일한 기능을 가지고 있어야 합니다.

실제 서버 그룹 용 하드웨어를 선택 및 설정하기 전, 사용할 세 가지 로드 밸런서 추가 기능 토폴로지를 결정해야 합니다.

3.1. NAT 로드 밸런서 추가 기능 네트워크

NAT 토폴로지는 기존 하드웨어를 사용하는 데에 있어서 광대한 사용 범위를 허용하지만 로드 밸런서 추가 기능 라우터를 통해 모든 패킷이 풀로 나가고 들어오기 때문에 거대한 부하량을 처리하는 데에는 기능이 제한되어 있습니다.

네트워크 레이아웃

NAT 라우팅을 사용한 로드 밸런서 추가 기능 용 토플로지는 클러스터가 공개 네트워크로 하나의 액세스 지점만을 필요로하기 때문에 네트워크 레이아웃에서 설정하는 것이 가장 간편합니다. 실제 서버는 모든 요청을 LVS 라우터를 통해 전달하므로 실제 서버의 사설 네트워크 상에 있게 됩니다.

하드웨어

NAT 토플로지는 실제 서버가 Linux 컴퓨터가 아니어도 올바르게 작동하게 하므로 하드웨어와 관련하여 가장 유연한 방법입니다. NAT 토플로지에서 각각의 실제 서버는 LVS 라우터에만 응답하기 때문에 하나의 NIC만을 필요로 합니다. 반면 LVS 라우터는 두 개의 네트워크 사이에서 소통량을 라우트 하기 위해 각각 두개의 NIC를 필요로 합니다. 이러한 토플로지에서는 LVS 라우터가 병목 지점이 될 수 있습니다. 기가비트 이더넷 NIC를 각각의 LVS 라우터에서 사용하면, LVS 라우터가 처리할 수 있는 대역폭을 증가시킬 수 있습니다. 기가비트 이더넷을 LVS 라우터에 사용할 경우, 실제 서버를 LVS 라우터로 연결하는 모든 스위치들은 부하량을 효과적으로 처리하기 위해 최소 두개의 기가비트 이더넷 포트가 있어야 합니다.

소프트웨어

몇몇 NAT 토폴로지 설정에서는 **iptables**을 사용해야 하기 때문에, **Piranha Configuration Tool** 바깥에서 소프트웨어 설정을 꽤 많이 해야 할 수도 있습니다. 특히, 방화벽 표지 기능을 사용하는 FTP 서비스 등은 요청을 올바르게 라우트하기 위해 LVS 라우터에 수동으로 추가 설정을 해야 합니다.

3.1.1. NAT를 사용한 로드 밸런서 추가 기능 용 네트워크 인터페이스 설정

NAT로 로드 밸런서 추가 기능을 설정하기 위해, 우선 LVS 라우터에 있는 공개 네트워크 및 개인 네트워크 용 네트워크 인터페이스를 설정해야 합니다. 예에서, LVS 라우터의 공개 인터페이스 (**eth0**)는 192.168.26/24 네트워크 상에 있게 되며 (물론 이는 라우터 가능한 IP가 아니지만, LVS 라우터 앞에 방화벽이 있다고 가정합시다) 실제 서버에 연결되어 있는 사설 인터페이스(**eth1**)는 10.11.12/24 네트워크 상에 있게 됩니다.

활성 또는 *주* LVS 라우터 노드 상에서, 공개 인터페이스의 네트워크 스크립트, **/etc/sysconfig/network-scripts/ifcfg-eth0**는 다음과 같이 나타납니다:

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

LVS 라우터 상의 사설 NAT 인터페이스 용 **/etc/sysconfig/network-scripts/ifcfg-eth1** 파일은 다음과 같이 나타납니다:

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

예에서 LVS 라우터의 공개 인터페이스 용 VIP는 192.168.26.10이 되고 NAT 또는 개인 인터페이스 용 VIP는 10.11.12.10이 됩니다. 따라서, 실제 서버가 NAT 인터페이스 용 VIP로 요청을 라우트하는 것이 필수적입니다.

중요

이 절에 있는 이더넷 인터페이스 설정 예는 LVS 라우터의 실제 IP 주소에 대한 것이지 유동 IP 주소에 대한 것이 *아닙니다*. 4.4절. “GLOBAL SETTINGS” 및4.6.1절. “VIRTUAL SERVER 하위 섹션”에서 볼 수 있듯이, 공개 및 사설 유동 IP 주소를 설정하려면 관리자는 **Piranha Configuration Tool**를 사용해야 합니다.

주 LVS 라우터 노드의 네트워크 인터페이스들을 설정한 후, 백업 LVS 라우터의 실제 네트워크 인터페이스를 설정합니다 — 어떤 IP 주소도 네트워크 상의 다른 IP 주소와 충돌하지 않도록 조심하십시오.

중요

백업 노드에 있는 인터페이스는 주 노드에 있는 인터페이스와 같은 네트워크를 사용해야 합니다. 예를 들어, eth0가 주 노드 상의 공개 네트워크에 연결되어 있을 경우, 이는 백업 노드 상의 공개 네트워크에도 연결되어 있어야 합니다.

3.1.2. 실제 서버에서 라우팅

NAT 토폴로지에서 실제 서버 네트워크 인터페이스 설정 시 기억해 두어야 할 가장 중요한 사항은 LVS 라우터의 NAT 유동 IP 주소 용 게이트웨이를 설정해야 하는 것입니다. 예에서, 이 주소는 10.11.12.10으로 되어 있습니다.

알림

네트워크 인터페이스가 실제 서버에서 활성화되면 컴퓨터는 핑을 할 수 없게 되거나 다른 방법으로 공개 네트워크에 연결하는 것이 일반적입니다. 하지만, LVS 라우터의 개인 인터페이스에 대해서는 실제 IP를 핑할 수 있습니다. 이 경우, 10.11.12.8이 이에 해당합니다.

실제 서버의 **/etc/sysconfig/network-scripts/ifcfg-eth0** 파일은 다음과 같이 나타납니다:

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

경고

실제 서버의 **GATEWAY=** 행에 하나 이상의 네트워크 인터페이스가 설정되어 있을 경우, 첫번째로 나타나는 것이 게이트웨이가 됩니다. 따라서 **eth0**와 **eth1**이 설정되어 있고 **eth1**이 로드 밸런서 추가 기능에 사용될 경우, 실제 서버가 요청을 올바르게 라우트하지 못할 수 도 있습니다. **/etc/sysconfig/network-scripts/** 디렉토리에 있는 네트워크 스크립트에서 **ONBOOT=no**를 설정하거나, 처음으로 나타나는 인터페이스에 게이트웨이가 올바르게 설정되어 있는 지를 확인하여 관련 없는 네트워크 인터페이스를 비활성화시키는 것이 좋습니다.

3.1.3. LVS 라우터에서 NAT 라우팅 활성화

각각의 클러스터된 서비스가, 예를 들어 포트 80에 있는 HTTP와 같이, 하나의 포트만을 사용하는 단순한 NAT 로드 밸런서 추가 기능 설정에서는 관리자는 외부와 실제 서버 사이에서 요청이 올바르게 라우트되도록 LVS 라우터에 있는 패킷 포워딩만을 활성화하면 됩니다. 패킷 포워딩을 활성화하는 방법에 대한 내용은 2.5절. “패킷 포워딩 활성화 ”에서 참조하기 바랍니다. 하지만 사용자 세션 동안 클러스터된 서비스가 둘 이상의 포트가 동일한 실제 서버로 가야할 필요가 있는 경우, 다른 설정이 필요합니다. 방화벽 표시 기능을 사용한 다중 포트 서비스 생성 방법에 대한 내용은 3.4절. “다중 포트 서비스 및 로드 밸런서 추가 기능 ”에서 참조하시기 바랍니다.

일단 LVS 라우터에서 포워딩이 활성화되고, 실제 서버가 설정되어, 클러스터된 서비스가 실행되면, 4장. *Piranha Configuration Tool*를 사용하여 로드 밸런서 추가 기능 (Load Balancer Add-On) 설정하기 과 같이 로드 밸런서 추가 기능을 설정하기 위해 **Piranha Configuration Tool**를 사용합니다.

경고

수동으로 네트워크 스크립트를 편집하거나 네트워크 설정 도구를 사용하여 **eth0:1** 또는 **eth1:1**에 대한 유동 IP를 설정하지 마십시오. 대신 4.4절. “**GLOBAL SETTINGS**” 및 4.6.1절. “**VIRTUAL SERVER** 하위 섹션”과 같이 **Piranha Configuration Tool**를 사용하셔야 합니다.

설정을 마치면, 4.8절. “로드 밸런서 추가 기능 시작 ”에서 보여주듯 **pulse** 서비스를 시작합니다. **pulse**가 실행되면, 활성 LVS 라우터는 실제 서버 풀로 요청을 라우팅할 것입니다.

3.2. 직접 라우팅을 통한 로드 밸런서 추가 기능

1.4.2절. “직접 라우팅 ”에서 언급하였듯이, 직접 라우팅은 실제 서버가 패킷을 요청한 사용자에게 직접 라우트하도록 합니다(LVS 라우터를 통해 전달하지 않습니다). 직접 라우팅에서는 실제 서버가 LVS 라우터가 있는 네트워크 세그먼트에 물리적으로 연결되어, 나가는 패킷을 직접 처리하고, 라우팅할 수 있어야 합니다.

네트워크 레이아웃

직접 라우팅 로드 밸런서 추가 기능 설정에서 LVS 라우터는 들어오는 요청을 받아 이를 처리하기 위해 실제 서버로 올바르게 라우트해야 합니다. 그 후 실제 서버는 이 응답을 클라이언트에게 *직접* 라우트합니다. 예를 들어, 클라이언트가 인터넷 상에서 LVS 라우터를 통해 실제 서버로 패킷을 전송할 경우, 실제 서버는 인터넷을 통해 클라이언트에게로 직접 도달할 수 있어야 합니다. 인터넷으

로 패킷을 전달하기 위해 실제 서버에 대한 게이트웨이를 설정하여 이렇게 할 수 있습니다. 서버 풀에 있는 각각의 실제 서버는 최대 처리량 및 확장성을 허용하기 위해, 자신만의 분리된 게이트웨이 (각 게이트웨이는 인터넷에 대해 자신만의 연결을 가짐)를 사용할 수 도 있습니다. 하지만, 전형적인 로드 밸런서 추가 기능 설정의 경우 실제 서버는 하나의 게이트웨이 (즉 하나의 네트워크 연결)를 통해 통신할 수 있습니다.

중요

불필요하게 복잡한 설정을 필요로 할 뿐 아니라, LVS 라우터에 네트워크 부하량을 추가함으로써 NAT 라우팅에서 발생했던 네트워크 병목 현상을 다시 발생시킬 수 있기 때문에, 실제 서버 용 게이트웨이로서 LVS 라우터를 사용하는 것은 *권장하지 않습니다*.

하드웨어

직접 라우팅을 사용한 로드 밸런서 추가 기능 시스템의 하드웨어 필요 요건은 기타 다른 로드 밸런서 추가 기능 토폴로지와 비슷합니다. 들어오는 요청을 처리하고 실제 서버에 필요한 로드 밸런싱을 실행하기 위해 LVS 라우터가 Red Hat Enterprise Linux에서 실행되고 있어야 하는 반면, 올바른 작동을 위해 실제 서버가 Linux를 사용하는 컴퓨터에서 실행될 필요는 없습니다. LVS 라우터에는 (백업 라우터가 있는 지의 여부에 따라) 각각 하나 또는 두개의 NIC가 있어야 합니다. 설정을 쉽게 하기 위해 두 개의 NIC를 사용하여 트래픽을 분할할 수 있습니다 — 들어오는 요청은 하나의 NIC에 의해 처리되고 다른 NIC에서는 실제 서버로 패킷을 라우트합니다.

실제 서버가 LVS 라우터를 무시하고 클라이언트에게 나가는 패킷을 직접 전달하기 때문에 인터넷으로의 게이트웨이가 있어야 합니다. 성능 및 가용성을 최대화하기 위해 각각의 실제 서버는 클라이언트가 연결되어 있는 네트워크(인터넷 또는 인트라넷)에 자체적으로 접속되는 분리된 게이트웨이로 연결될 수 있습니다.

소프트웨어

특히 직접 라우팅을 통해 로드 밸런서 추가 기능을 사용할 때 ARP 문제가 있는 경우, **Piranha Configuration Tool** 외부에서 관리자가 실행해야 할 설정 사항이 있습니다. 보다 자세한 내용은 3.2.1절. “직접 라우팅 및 **arptables_jf**” 또는 3.2.2절. “직접 라우팅 및 **iptables**”에서 참조하시기 바랍니다.

3.2.1. 직접 라우팅 및 arptables_jf

arptables_jf를 사용하여 직접 라우팅을 설정하기 위해, 각각의 실제 서버는 가상 IP 주소를 설정하여 직접 패킷을 라우트할 수 있어야 합니다. VIP에 필요한 ARP 요청은 실제 서버에 의해 전적으로 무시되며, 다른 경우 전달될 수도 있었을 VIP가 들어있는 ARP 패킷들은 VIP 가 아닌 실제 서버의 IP를 포함도록 조작됩니다.

arptables_jf 방식을 사용해, 어플리케이션은 실제 서버가 사용하고 있는 각각의 개별적 VIP나 포트를 바인드할 수 있습니다. 예를 들어, **arptables_jf** 방식은 여러 Apache HTTP Server 인스턴스가 시스템 상의 다른 VIP로 바인드될 수 있도록 허용합니다. 또한, **iptables** 방식을 사용하는 것 보다는 **arptables_jf**를 사용하는 것이 상당한 성능상의 잇점이 있습니다.

하지만, **arptables_jf** 방식을 사용하면 VIP는 Red Hat Enterprise Linux 시스템 설정 도구를 사용하여 부팅시 시작되도록 설정될 수 없습니다.

각각의 가상 IP 주소에 해당하는 ARP 요청을 무시하기 위해 각각의 실제 서버를 설정하려면, 다음과 같은 절차를 수행합니다:

1. 실제 서버에서 각각의 가상 IP 주소에 해당하는 ARP 테이블 항목을 생성합니다 (real_ip는 실제 서버

와 통신하기 위해 director가 사용하는 IP입니다; 주로 이는 **eth0**로 바인드된 IP입니다):

```
arptables -A IN -d <virtual_ip> -j DROP
arptables -A OUT -d <virtual_ip> -j mangle --mangle-ip-s <real_ip>
```

이렇게 하면, 실제 서버는 가상 IP 주소에 대한 모든 ARP 요청을 무시하게 되며, 가상 IP가 들어있는 나가는 ARP 응답을 변경해서 서버의 실제 IP를 포함시킵니다. VIP 주소에 대한 ARP 요청을 응답할 수 있는 유일한 노드는 현재 활성화된 LVS 노드 뿐입니다.

2. 각각의 실제 서버에서 이러한 작업을 완료한 후, 다음과 같은 명령을 사용하여 ARP 테이블 항목을 저장합니다:

   service arptables_jf save

   chkconfig --level 2345 arptables_jf on

   chkconfig 명령으로 부팅시 — 네트워크가 시작되기 전 arptables 설정을 시스템으로 다시 불러올 수 있습니다.

3. IP 알리아스를 생성하기 위해 **ifconfig**를 사용하여 모든 실제 서버에 있는 가상 IP 주소를 설정합니다. 예를 들면:

   ```
   # ifconfig eth0:1 192.168.76.24 netmask 255.255.252.0 broadcast
   192.168.79.255 up
   ```

   또는 **iproute2** 유틸리티의 **ip** 명령을 사용합니다 예:

   ```
   # ip addr add 192.168.76.24 dev eth0
   ```

   이전에 언급하였듯이 가상 IP 주소는 Red Hat 시스템 설정 도구를 사용하여 부팅시 시작되도록 설정할 수 없습니다. 이 문제를 해결하는 방법 중 하나는 이 명령어들을 **/etc/rc.d/rc.local**에 포함시키는 것입니다.

4. 직접 라우팅 용 Piranha를 설정합니다. 보다 자세한 내용은 *4장. Piranha Configuration Tool를 사용하여 로드 밸런서 추가 기능 (Load Balancer Add-On) 설정하기*에서 참조하십시오.

3.2.2. 직접 라우팅 및 iptables

iptables 방화벽 규칙을 생성하는 방법으로, 직접 라우팅 방식을 사용할 때 발생하는 ARP 문제를 해결할 수 있습니다. **iptables**를 사용하여 직접 라우팅을 설정하려면, 투명 프록시 (transparent proxy)를 생성하는 규칙을 추가하여 시스템에 VIP 주소가 없더라도 실제 서버가 VIP 주소로 보내는 패킷을 서비스하도록 해야 합니다.

iptables 방식은 설정에 있어서 **arptables_jf** 방식과 비슷합니다. 이 방식은 가상 IP 주소가 활성 LVS director에만 있게 되므로, LVS ARP 문제가 발생하는 것을 전적으로 막을 수 있습니다.

하지만, **iptables** 방식을 **arptables_jf** 방식과 비교하면, 모든 패킷을 포워딩/마스커레이딩하는데 필요한 오버헤드로 인해, 성능면에서 문제가 발생합니다.

iptables 방식을 사용하면 포트를 재사용할 수 없습니다. 예를 들어, 포트 80으로 바인드된 두개의 분리된 Apache HTTP Server 서비스를 실행할 수 없습니다. 왜냐하면 이러한 서비스는 가상 IP 주소 대신 ***INADDR_ANY***로 바인드되어야 하기 때문입니다.

iptables 방식을 사용하여 직접 라우팅을 설정하려면 다음과 같은 절차를 수행합니다:

1. 각각의 실제 서버에서, 사용하고자 하는 모든 VIP, 포트, 프로토콜 (TCP or UDP) 조합에 대해 다음과 같은 명령을 실행합니다:

   iptables -t nat -A PREROUTING -p <tcp|udp> -d <vip> --dport <port> -j REDIRECT

이 명령으로 실제 서버는 VIP로 목적지가 지정된 패킷을 주어진 곳으로 포트하게 합니다.

2. 각 실제 서버에 설정 사항을 저장합니다:

```
# service iptables save
# chkconfig --level 2345 iptables on
```

위의 명령은 부팅시 — 네트워크가 시작되기 전 시스템이 **iptables** 설정을 다시 읽어오게 합니다.

3.3. 설정을 함께 저장하기

앞에서 설명한 라우팅 방식 중 사용할 것을 결정한 후, 하드웨어를 네트워크 상에 연결해야 합니다.

중요

LVS 라우터에 있는 어댑터 장치는 동일한 네트워크를 액세스하도록 설정해야 합니다. 예를 들어 **eth0**가 공개 네트워크에 연결되어 있고 **eth1**이 사설 네트워크에 연결되어 있다면, 백업 LVS 라우터에 있는 동일한 장치는 동일한 네트워크로 연결해야만 합니다.
부팅시 나타나는 첫 번째 인터페이스에 있는 게이트웨이는 라우팅 테이블에 추가되며 다른 인터페이스에 있는 다음의 게이트는 무시됩니다. 이는 실제 서버를 설정할 때 고려해야 할 아주 중요한 사항입니다.

물리적으로 하드웨어를 연결한 후에, 주 LVS 라우터와 백업 LVS 라우터에 있는 네트워크 인터페이스를 설정합니다. **system-config-network**와 같은 그래픽 어플리케이션을 사용하거나 수동으로 네트워크 스크립트를 편집하여 설정할 수 있습니다. **system-config-network**를 사용하여 장치를 추가하는 방법에 대한 자세한 내용은 *Red Hat Enterprise Linux 활용 가이드*에 있는 *네트워크 설정* 장을 참조하시기 바랍니다. 이 장의 나머지 부분에서는 네트워크 인터페이스를 수동으로 변경하는 예와, **Piranha Configuration Tool**를 통해 변경하는 예를 다룹니다.

3.3.1. 일반적인 로드 밸런서 추가 기능 네트워킹 조언

Piranha Configuration Tool를 사용하여 로드 밸런서 추가 기능을 설정하기 전에 LVS 라우터에 있는 공개 및 사설 네트워크에 해당하는 실제 IP 주소를 설정하십시오. 각각의 토폴로지에 대한 섹션에서 네트워크 주소의 예가 있지만, 실제 네트워크 주소가 필요합니다. 아래에는 네트워크 인터페이스를 활성화하거나 상태를 확인하기 위해 사용되는 유용한 명령이 있습니다.

실제 네트워크 인터페이스 활성화

실제 네트워크 인터페이스를 활성화하기 위해 root로 다음과 같은 명령을 사용합니다, 여기서 ***N***은 해당 인터페이스 번호로 대체합니다.(**eth0** 및 **eth1**)

```
/sbin/ifup ethN
```

경고

Piranha Configuration Tool를 사용하여 설정한 유동 IP 주소(**eth0:1** 또는 **eth1:1**)를 활성화시키기 위해 **ifup** 스크립트를 사용하지 *마십시오*. 대신 **pulse**를 시작하기 위해 **service** 명령을 사용합니다(자세한 내용은 4.8절. “로드 밸런서 추가 기능 시작 ”에서 참조).

실제 네트워크 인터페이스 비활성화

실제 네트워크 인터페이스를 비활성화시키기 위해 root로 다음과 같은 명령을 사용합니다, 여기서 ***N***은 해당 인터페이스 번호로 대체합니다. (**eth0** 및 **eth1**)

```
/sbin/ifdown ethN
```

네트워크 인터페이스 상태 확인

어떤 네트워크 인터페이스가 활성화되어 있는 지를 확인하려면 다음과 같은 명령을 사용합니다:

```
/sbin/ifconfig
```

컴퓨터에 해당하는 라우팅 테이블을 보려면, 다음과 같은 명령을 실행합니다:

```
/sbin/route
```

3.4. 다중 포트 서비스 및 로드 밸런서 추가 기능

어떤 토폴로지에서건 LVS 라우터는 다중 포트 로드 밸런서 추가 기능 서비스를 생성할 때 추가 설정을 해야 합니다. HTTP (80 포트)와 HTTPS (443 포트)와 같이 서로 다르지만 관련된 프로토콜을 묶거나, FTP와 같은 다중 포트 프로토콜 서비스에서 로드 밸런서 추가 기능을 사용하는 경우, 방화벽 표시 기능으로 다중 포트 서비스를 인위적으로 생성할 수 있습니다. 이 경우, LVS 라우터는 방화벽 표시 기능을 사용하여, 목적지 포트는 다르지만 동일한 방화벽 표시를 갖는 패킷을 인식해서, 동일하게 처리해야 합니다. 또한 지속성 기능과 결합하여 사용할 때, 방화벽 표시 기능은 클라이언트 컴퓨터에서의 연결을 확인하고, 지속성 매개 변수에 의해 지정된 시간 이내로 연결이 지속될 경우, 같은 호스트로 라우트합니다. 가상 서버로 지속성 기능을 지정하는 방법에 대한 자세한 내용은 4.6.1절. “**VIRTUAL SERVER** 하위 섹션”에서 참조하시기 바랍니다.

실제 서버에서의 로드 밸런스를 유지하기 위해 사용된 메카니즘 — IPVS —은 패킷에 지정된 방화벽 표시 기능을 인식할 수 있으나, 그 자체로 방화벽 표지를 지정할 수는 없습니다. 방화벽 표지를 *지정하는* 작업은 **Piranha Configuration Tool**의 밖에서 네트워크 패킷 필터인 **iptables**에 의해 실행되어야 합니다.

3.4.1. 방화벽 표지 지정

특정 포트로 목적지가 지정된 패킷에 방화벽 표지를 지정하기 위해 관리자는 **iptables**를 사용해야 합니다.

다음 섹션에서는 HTTP 및 HTTPS를 묶는 방법에 대한 예를 보여줍니다; 하지만, FTP는 또 다른 방식으로 클러스터되는 다중 포트 프로토콜입니다. 로드 밸런서 추가 기능이 FTP 서비스에 대해 사용될 경우 설정 방법에 대한 자세한 내용은 3.5절. “FTP 설정 ”에서 참조하시기 바랍니다.

방화벽 표지를 지정할 때 기억해 두어야 할 기본적인 규칙은 **Piranha Configuration Tool**에서 방화벽 표지를 사용하는 모든 프로토콜에 대해서, 네트워크 패킷에 표지를 지정하는 적당한 **iptables** 규칙이 있어야 한다는 것입니다.

네트워크 패킷 필터 규칙을 만들기 전에 이미 있는 규칙이 없는지 확인합니다. 이를 위해 쉘 프롬프트를 열고 root로 로그인하여 다음을 실행합니다:

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

iptables가 실행중이 아니라면, 바로 프롬프트가 다시 나타납니다.

iptables가 활성화되어 있다면, 이 명령은 규칙 모음을 보여줍니다. 규칙이 존재한다면 다음과 같은 명령을 입력합니다:

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

기존에 있는 규칙이 중요한 것일 경우, **/etc/sysconfig/iptables**의 내용을 확인해서, 저장해 두어야할 규칙을 안전한 장소에 복사해 둡니다.

아래에 있는 것은 포트 80과 443에서 유동 IP 주소 ***n.n.n.n***로 목적지가 지정된 들어오는 트래픽에 같은 방화벽 표지 80을 지정하는 규칙입니다.

```
/sbin/modprobe ip_tables
```

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

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

공개 네트워크 인터페이스에 VIP를 지정하는 방법에 대한 내용은 4.6.1절. “**VIRTUAL SERVER** 하위 섹션”에서 참조하시기 바랍니다. 또한, 처음으로 규칙을 실행하기 전에, root로 로그인하여 **iptables**에 해당하는 모듈을 읽어와야 한다는 것을 명심하십시오.

위의 **iptables** 명령에서, ***n.n.n.n***은 HTTP 및 HTTPS 가상 서버에 해당하는 유동 IP로 대체해야 합니다. 이 명령은 포트 상의 VIP에서 인식되는 모든 트래픽에 대해 80이라는 방화벽 표지를 지정하는 효과를 갖게 되고, 이제 80이라는 방화벽 표지는 IPVS에 의해 인식되어 올바르게 포워드됩니다.

경고

위의 명령은 즉시 영향을 미치지만 시스템 재부팅시 복구되지는 않습니다. 네트워크 패킷 필터 설정이 재부팅시 복구되도록 하는 것은 3.6절. “네트워크 패킷 필터 설정 저장 ”를 참조하십시오.

3.5. FTP 설정

파일 전송 프로토콜(File Transport Protocol,FTP)은 로드 밸런서 추가 기능 환경 구축의 문제점들을 명확히 보여주는 오래되고 복잡한 다중 포트 프로토콜입니다. 이러한 본질적 어려움을 이해하기 위해서는 FTP 작동 방법에 대한 핵심 사항을 파악하고 있어야 합니다.

3.5.1. FTP 작동 원리

대부분의 다른 서버 클라이언트 관계에서는, 클라이언트 컴퓨터가 특정 포트에 있는 서버로 연결을 열면 서버는 그 포트를 통해 클라이언트에 응답합니다. FTP 클라이언트가 FTP 서버에 연결되면, 클라이언트는 FTP 제어 포트 21로의 연결을 열게 됩니다. 그 후 *클라이언트*는 FTP *서버*에게 *수동* 또는 *능동* 연결 모드로 할 지를 지정합니다. 클라이언트에 의해 지정된 연결 모드는 서버 응답 방법 및 어떤 포트에서 트랜잭션을 실행할 지를 결정합니다.

데이터 연결에는 다음과 같이 두 가지 유형이 있습니다:

능동 연결 모드

능동 연결 모드를 사용할 경우, *서버*는 클라이언트 컴퓨터에 있는 포트 20에서 상위 범위의 포트로 클라이언트로의 데이터 연결을 오픈합니다. 서버로 부터 모든 데이터는 이 연결을 통해 전달됩니다.

수동 연결 모드

수동 연결 모드를 사용할 경우, *클라이언트*는 FTP 서버에게 10,000 이상의 포트에서 수동 연결 포트를 구성할 것을 요청합니다. 서버는 이러한 특정 세션을 위해 높은 번호의 포트를 바인드하고, 클라이언트에게 바인되한 포트 번호를 전달합니다. 클라이언트는 데이터 연결을 위해 새로 바인드된

포트를 엽니다. 클라이언트가 요청하는 데이터 요청은 별개의 데이터 연결로 분리됩니다. 현재 대부분의 FTP 클라이언트는 서버에서 데이터를 요청할 때 수동 연결 모드를 사용합니다.

알림

서버가 아니고 *클라이언트*가 연결 모드 유형을 지정합니다. 이는 FTP를 효과적으로 클러스터하기 위해서는 수동 및 능동 연결 모드를 모두 처리할 수 있도록 LVS 라우터를 설정해야 한다는 것을 의미합니다.
FTP 클라이언트/서버 관계는 **Piranha Configuration Tool**와 IPVS가 알지 못하는 대량의 포트를 열 수 있는 가능성도 있습니다.

3.5.2. 로드 밸런서 추가 기능 라우팅에 미치는 영향

IPVS 패킷 포워딩은 포트 번호나 방화벽 표지에 의해 구별되는 클러스터로 들어오고 나가는 모든 연결 만을 허용합니다. 클러스터 외부에 있는 클라이언트가 IPVS가 사용하도록 설정되지 않은 포트를 열 경우, 그 연결은 종료됩니다. 마찬가지로, 실제 서버가 IPVS가 알지 못하는 포트를 외부 인터넷에 대해 열려고 하는 경우, 그 연결도 종료됩니다. 이는 인터넷 상의 FTP 클라이언트에 들어오는 *모든* 연결에는 *반드시* 동일한 방화벽 표지가 지정되어 있어야 하고, FTP 서버에서 나가는 모든 연결은 *반드시* 네트워크 패킷 필터링 규칙을 사용하여 인터넷으로 올바르게 포워드되어야 한다는 것을 의미합니다.

3.5.3. 네트워크 패킷 필터 규칙 생성

FTP 서비스에 필요한 **iptables** 규칙을 지정하기 전에 3.4.1절. “방화벽 표지 지정 ”에서 다중 포트 서비스와 기존의 네트워크 패킷 필터링 규칙을 확인하는 기법에 대한 정보를 다시 살펴보십시오.

아래에는 FTP 트래픽으로 21이라는 동일한 방화벽 표지를 지정한 규칙이 있습니다. 이러한 규칙이 올바르게 작동하게 하려면, **Piranha Configuration Tool**의 **VIRTUAL SERVER** 하위 섹션안에 **Firewall Mark** 란에 **21**을 입력하여 포트 21에 해당하는 가상 서버를 설정해야 합니다. 보다 자세한 내용은 4.6.1절. “VIRTUAL SERVER 하위 섹션”에서 참조하시기 바랍니다.

3.5.3.1. 능동 연결 규칙

능동 연결에 해당하는 규칙에서는 커널이 포트 20 상의 *내부* 유동 IP 주소로 들어오는 연결을 허용하고 포워드하게 합니다.

다음의 **iptables** 명령에서는 LVS 라우터가 IPVS가 알지 못하는 실제 서버에서 나가는 연결을 허용하게 합니다:

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

iptables 명령에서 ***n.n.n***은 **Piranha Configuration Tool**의 **GLOBAL SETTINGS** 패널에서 지정한 NAT 인터페이스의 내부 네트워크 인터페이스의 유동 IP 값의 처음 세 값으로 대체합니다.

3.5.3.2. 수동 연결 규칙

수동 연결 규칙에서는 인터넷에서 서비스의 유동 IP로 들어오는 넓은 범위의 포트에 대한 연결에 같은 방화벽 표지를 지정합니다— 10,000부터 20,000.

경고

수동 연결에 대해 포트 범위에 제한을 둘 경우, VSFTP 서버를 설정하여 포트 범위를 일치시켜야 합니다. 이를 위해 **/etc/vsftpd.conf**에 다음과 같은 행을 추가합니다:
pasv_min_port=10000
pasv_max_port=20000
수동 FTP 연결을 위해 클라이언트에게 보여지는 서버 주소를 관리해야 합니다. 로드 밸런서 추가 기능 시스템에 라우트된 NAT에서 **/etc/vsftpd.conf**에 다음과 같은 행을 추가하여 연결시 클라이언트에게 보여지는 실제 서버 IP 주소를 VIP로 대체합니다. 예:
pasv_address=*n.n.n.n*
n.n.n.n을 LVS 시스템의 VIP 주소로 대체합니다.
다른 FTP 서버를 설정하는 것은, 각 서버의 관련 문서를 참조하시기 바랍니다.

포트 범위는 대부분의 경우를 포용할 수 있을 만큼 광범위해야 합니다; 하지만, 아래에 있는 명령에서 **10000:20000**을 **1024:65535**로 변경하면 사용 가능한 모든 안전하지 않은 포트를 포함하도록 범위를 늘릴 수 있습니다.

다음의 **iptables** 명령은 포트 상의 유동 IP에서 인식한 모든 소통량에 대해 21이라는 방화벽 표지를 지정하는 효과를 갖습니다. 이는 다시 IPVS에 의해 인식되어 바르게 포워드됩니다:

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

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

iptables 명령에서 ***n.n.n.n***을 **Piranha Configuration Tool**의 **VIRTUAL SERVER** 하위 섹션에서 지정한 FTP 가상 서버의 유동 IP로 대체합니다.

경고

위의 명령은 즉시 영향을 미치지만 시스템 재부팅 후에는 복구되지 않습니다. 네트워크 패킷 필터 설정이 재부팅시 복구되게 하는 것은 3.6절. “네트워크 패킷 필터 설정 저장 ”를 참조하십시오.

마지막으로 올바른 서비스가 알맞은 런레벨에서 실행되도록 설정되었는지를 확인해야 합니다. 보다 자세한 내용은 2.1절. “LVS 라우터 상의 설정 서비스 ”에서 참조하십시오.

3.6. 네트워크 패킷 필터 설정 저장

네트워크 패킷 필터를 적절하게 설정한 후에, 설정 사항을 저장하여 재부팅 후 복구되게 합니다. **iptables**의 경우 다음과 같은 명령을 사용합니다:

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

이렇게 하면 **/etc/sysconfig/iptables**에 설정 사항을 저장해서, 부팅 시 다시 불러오도록 합니다.

일단 이 파일이 작성한 다음에는, **/sbin/service** 명령을 사용해 **iptables**를 시작, 정지, 상태 확인(상태 스위치를 사용) 할 수 있습니다. **/sbin/service**는 자동으로 필요한 모듈을 읽어옵니다.
/sbin/service 명령을 사용하는 방법에 대해서는 2.3절. “**Piranha Configuration Tool** 서비스 시작 ”를 참조하십시오.

마지막으로, 적절한 서비스가 해당 런레벨에서 실행되도록 설정되었는 지를 확인해야 합니다. 이에 관한 자세한 내용은 2.1절. “LVS 라우터 상의 설정 서비스 ”를 참조하십시오.

다음에서는 LVS 라우터를 설정하기 위해 **Piranha Configuration Tool**를 사용하는 방법과 로드 밸런서 추가 기능을 활성화시키기 위해 필요한 절차에 대해 설명합니다.

4장. Piranha Configuration Tool를 사용하여 로드 밸런서 추가 기능 (Load Balancer Add-On) 설정하기

Piranha Configuration Tool는 로드 밸런서 추가 기능에 필요한 설정 파일 — **/etc/sysconfig/ha/lvs.cf**을 생성하기 위한 구조적 접근 방법을 제공합니다. 다음 부분에서는 **Piranha Configuration Tool**의 기본적인 실행 방법 및 설정 완료 후 로드 밸런서 추가 기능을 활성화하는 방법에 대해 설명합니다.

중요

로드 밸런서 추가 기능에 필요한 설정 파일은 엄격한 포맷 규칙을 따릅니다. **Piranha Configuration Tool**를 사용하는 것은 **lvs.cf**에서 구문 오류가 발생하지 않게 하는 가장 좋은 방법으로 결국 소프트웨어 장애가 발생하지 않게 됩니다.

4.1. 필요한 소프트웨어

Piranha Configuration Tool를 사용하기 위해서는 **piranha-gui** 서비스가 주요 LVS 라우터에서 실행되고 있어야 합니다. 로드 밸런서 추가 기능을 설정하려면 최소한 **links**와 같은 텍스트 전용 웹 브라우저 이상이 필요합니다. 다른 컴퓨터에서 LVS 라우터에 액세스 하려는 경우, root 사용자로 주요 LVS 라우터에 **ssh** 연결이 되어 있어야 합니다.

주요 LVS 라우터를 설정하는 동안 터미널 창에서 **ssh** 연결을 유지하는 것이 좋습니다. 이는 **pulse** 및 기타 다른 서비스 재시작, 네트워크 패킷 필터 설정, 문제 해결 도중 **/var/log/messages** 모니터링하는 데 안전한 방법을 제공합니다.

다음 부분에서는 **Piranha Configuration Tool**의 각각의 설정 페이지를 살펴보고 로드 밸런서 추가 기능 설정을 위해 이를 사용하는 방법에 대해 설명합니다.

4.2. Piranha Configuration Tool로 로그인

로드 밸런서 추가 기능 설정 시 항상 **Piranha Configuration Tool**로 주요 라우터를 설정하는 것부터 시작해야 합니다. 이렇게 하기 위해, 2.2절. “**Piranha Configuration Tool** 암호 설정 ”에서 설명하듯이 **piranha-gui** 서비스가 실행되고 있는 지와 관리자 암호가 설정되었는 지를 확인해야 합니다.

로컬에서 컴퓨터를 액세스할 경우, **Piranha Configuration Tool**에 액세스하기 위해 웹 브라우저에서 **http://localhost:3636**을 열 수 있습니다. 로컬이 아닌 경우, 서버의 호스트명이나 실제 IP 주소 뒤에 **:3636**을 입력합니다. 브라우저가 연결되면 그림 4.1. “The Welcome Panel”과 같은 화면이 나타납니다.

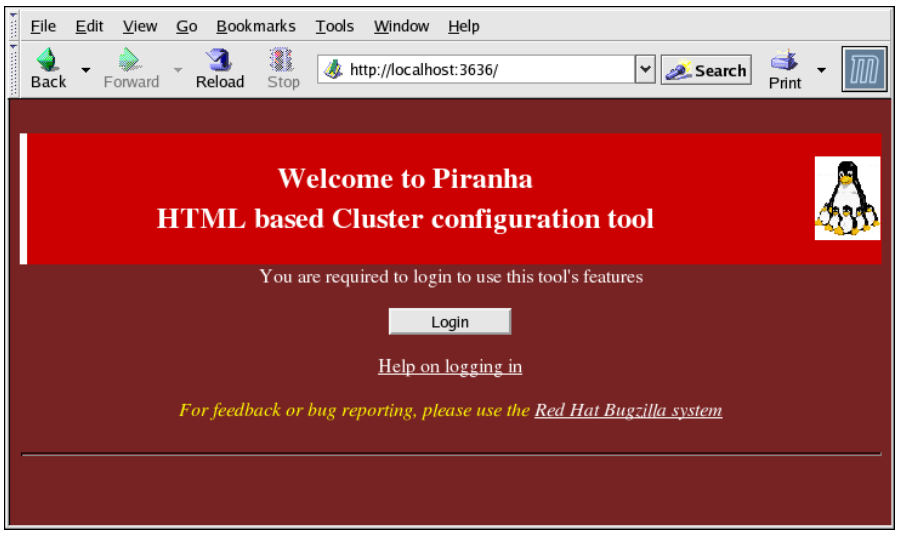

그림 4.1. The Welcome Panel

Login 버튼을 클릭하여 **Username**에 **piranha**를 입력하고 **Password** 란에는 여러분이 생성한 관리자 암호를 입력합니다.

Piranha Configuration Tool에는 네 개의 주요 화면, 즉 *패널*이 있으며, **Virtual Servers**에는 네 개의 *하위섹션*이 있습니다. **CONTROL/MONITORING** 패널은 로그인 화면 이후에 가장 처음으로 나타나는 패널입니다.

4.3. CONTROL/MONITORING

CONTROL/MONITORING 패널에서는 로드 밸런서 추가 기능의 제한적 런타임 상태를 보여줍니다. 여기서는 **pulse** 데몬, LVS 라우팅 테이블, LVS 스폰 **nanny** 프로세스의 상태를 보여줍니다.

알림

4.8절. “로드 밸런서 추가 기능 시작 ”에서 볼 수 있듯이 **CURRENT LVS ROUTING TABLE** 및 **CURRENT LVS PROCESSES** 란은 로드 밸런서 추가 기능을 실제로 시작할 때 까지 빈칸으로 남아 있게 됩니다.

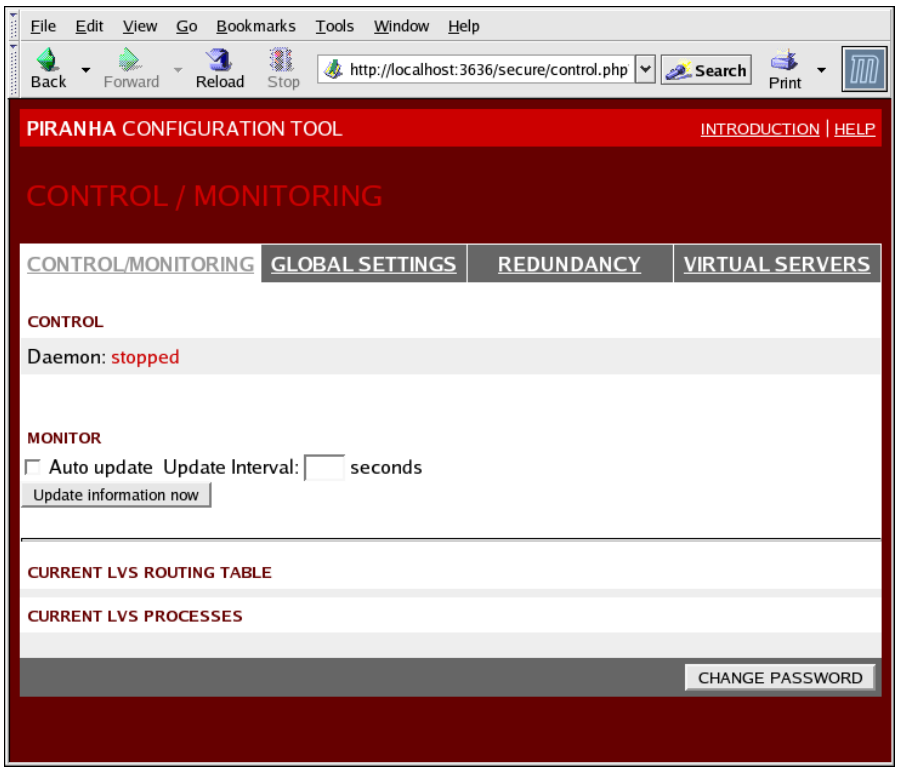

그림 4.2. CONTROL/MONITORING 패널

REAL SERVER

이 페이지에 있는 상태 보기는 사용자 설정 시간 간격으로 자동 업데이트될 수 있습니다. 이 기능을 활성화하려면 **Auto update**를 클릭하여 **Update frequency in seconds** (기본값은 10 초임) 텍스트 상자에서 원하는 업데이트 간격을 설정합니다.

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.

Auto update 업데이트 기능은 **Mozilla**에서와 같은 일부 브라우저에서는 작동하지 않습니다.

Update information now

이 버튼을 클릭하여 상태 정보를 수동으로 업데이트할 수 있습니다.

CHANGE PASSWORD

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

4.4. GLOBAL SETTINGS

GLOBAL SETTINGS 패널에서는 주요 LVS 라우터의 공개 및 사설 네트워크 인터페이스에 대한 네트워킹 정보를 지정합니다.

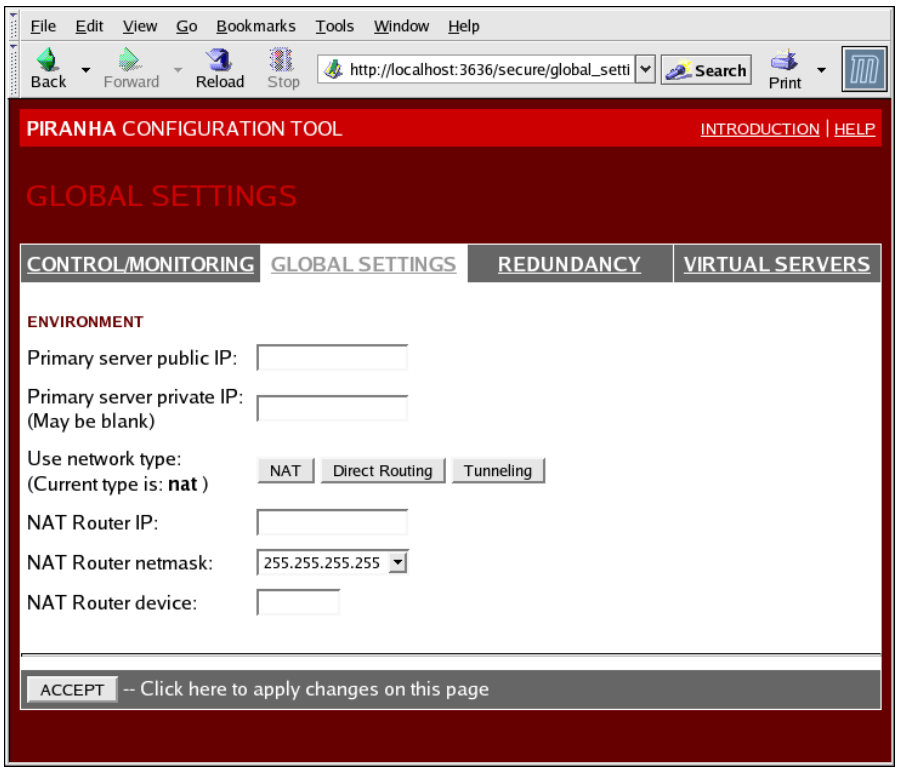

그림 4.3. GLOBAL SETTINGS 패널

패널의 위 절반에서는 주요 LVS 라우터의 공개 및 개인 네트워크 인터페이스를 설정합니다. 이 값은 3.1.1절. "NAT를 사용한 로드 밸런서 추가 기능 용 네트워크 인터페이스 설정 "에서 이미 설정되었습니다.

Primary server public IP

이 필드에는 주 LVS 노드에 해당하는 공개적으로 라우팅 가능한 실제 IP 주소를 입력합니다.

Primary server private IP

주 LVS 노드에 있는 다른 네트워크 인터페이스의 실제 IP 주소를 입력합니다. 이 주소는 백업 라우터의 다른 하트비트 채널용으로만 단독으로 사용되며 3.1.1절. "NAT를 사용한 로드 밸런서 추가 기능 용 네트워크 인터페이스 설정 "에서 지정한 실제 사설 IP 주소와는 관련이 없습니다. 이 란을 빈 칸으로 두어도 되지만, 빈 칸으로 둘 경우 백업 LVS 라우터로 사용할 다른 하트비트 채널이 없게 되어 단일 오류 지점(single point of failure,SPF)이 될 수 있습니다.

알림

모든 실제 서버와 LVS Director는 동일한 가상 IP 주소를 공유하고, 동일한 IP 라우트 설정을 공유해야만 하기 때문에, **Direct Routing** 설정을 위해 사설 IP 주소를 필요로 하지 않습니다.

알림

주 LVS 라우터의 사설 IP는, 이더넷 어댑터인지 시리얼 포트인지와는 상관 없이, TCP/IP를 사용가능한 모든 인터페이스에서 설정될 수 있습니다.

Use network type

NAT 버튼을 클릭하여 NAT 라우팅을 선택합니다.

Direct Routing 버튼을 클릭하여 직접 라우팅을 선택합니다.

다음의 세 부분에서는 사설 네트워크를 실제 서버와 연결하는 NAT 라우터의 가상 네트워크 인터페이스에 대해 다루고 있습니다. 이는 직접 라우팅 네트워크 유형에는 적용되지 *않습니다.*

NAT Router IP

사설 유동 IP를 입력합니다. 유동 IP는 실제 서버의 게이트웨이로 사용되어야 합니다.

NAT Router netmask

NAT 라우터의 유동 IP에 특정 넷마스크가 필요할 경우 드롭 다운 목록에서 선택합니다.

NAT Router device

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

알림

사설 네트워크에 연결된 이더넷 인터페이스로 NAT 유동 IP 주소를 알리아스해야만 합니다. 예를 들어, 사설 네트워크가 **eth1** 인터페이스에 있을 경우, **eth1:1**이 유동 IP 주소가 됩니다.

경고

이 페이지 작성을 완료한 후, **ACCEPT** 버튼을 클릭하여, 새 패널을 선택 시 변경 사항을 잃지 않도록 합니다.

4.5. REDUNDANCY

REDUNDANCY 패널에서는 백업 LVS 라우터 노드를 설정 하고, 다양한 하트비트 모니터링 옵션을 설정할 수 있습니다.

알림

처음으로 이 화면으로 이동했을 경우, "inactive" **Backup** 상태와 **ENABLE** 버튼이 나타납니다. 백업 LVS 라우터를 생성하려면 **ENABLE** 버튼을 클릭하여 화면을 그림 4.4. “**REDUNDANCY** 패널”과 같이 되게 합니다.

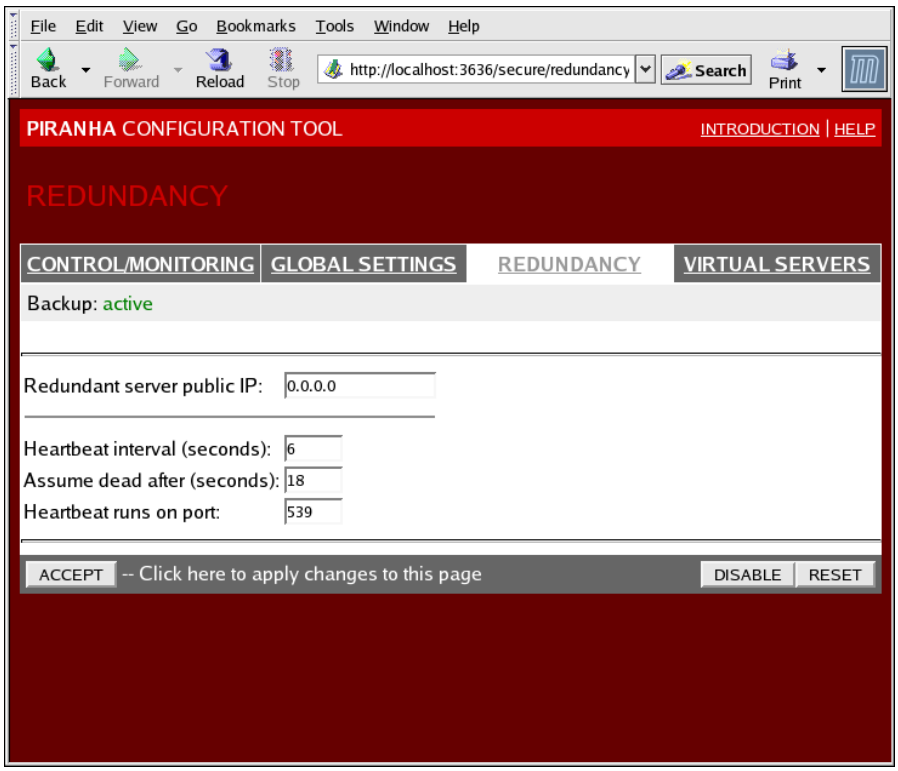

그림 4.4. REDUNDANCY 패널

Redundant server public IP

백업 LVS 라우터 노드용 공개 실제 IP 주소를 입력합니다.

Redundant server private IP

백업 노드의 사설 실제 IP 주소를 입력합니다.

Redundant server private IP라는 란이 나타나지 않을 경우, **GLOBAL SETTINGS** 패널로 돌아가서 **Primary server private IP** 주소를 입력하고 **ACCEPT** 버튼을 클릭합니다.

나머지 패널에서는 하트비트 채널을 설정하며, 문제 발생의 경우 백업 노드가 주 노드를 모니터하는데 사용됩니다.

Heartbeat Interval (seconds)

하트비트 사이의 간격을 초 단위로 설정합니다 — 즉, 백업 노드가 주 LVS 노드의 동작 상태를 점검하는 간격입니다.

Assume dead after (seconds)

지정한 시간(초)가 경과할 때까지 주 LVS 노드가 응답하지 않을 경우, 백업 LVS 라우터 노드가 장애 해결 조치를 시작하게 됩니다.

Heartbeat runs on port

주 LVS 노드와 하트비트 통신하는 포트를 설정합니다. 빈 칸으로 되어 있을 경우 기본값으로 539를 설정합니다.

경고

새 패널을 선택했을 때 이 패널에서 변경한 사항을 잃지 않으려면 **ACCEPT** 버튼을 클릭해야 합니다.

4.6. VIRTUAL SERVER

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

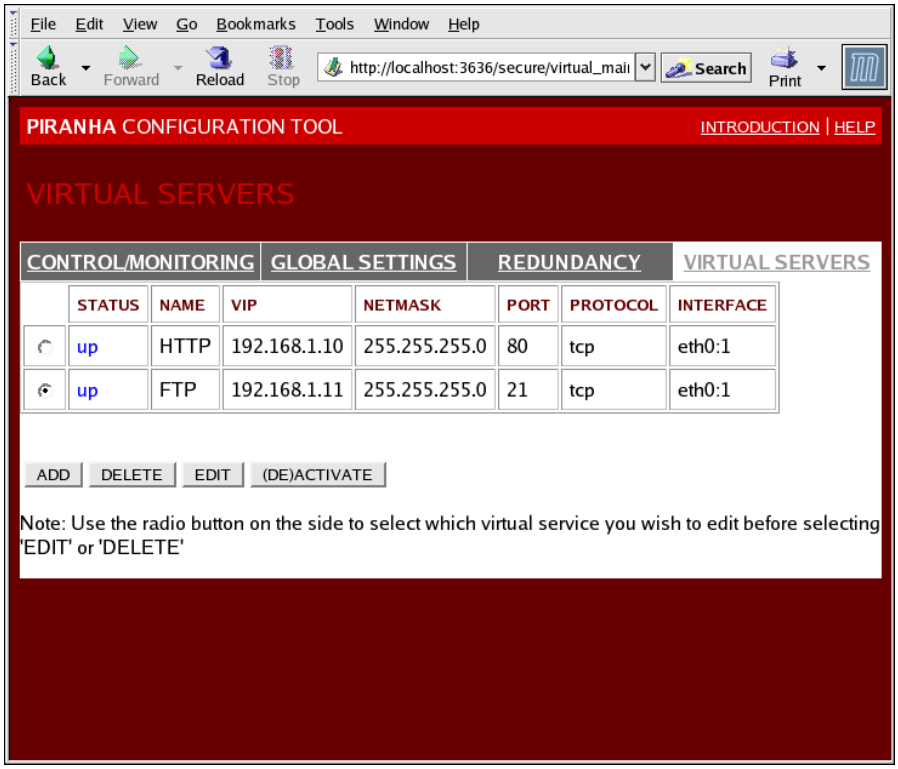

그림 4.5. VIRTUAL SERVERS 패널

VIRTUAL SERVERS 패널에 표시되는 각각의 서버는 다음의 스크린이나 또는 *하위 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 위해서는 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** 하위 섹션을 볼 수 있습니다.

4.6.1. VIRTUAL SERVER 하위 섹션

그림 4.6. "**VIRTUAL SERVER** 하위 섹션"에서 보여주듯이 **VIRTUAL SERVER** 하위 섹션 패널에서는 개별적인 가상 서버를 설정할 수 있습니다. 이러한 가상 서버와 연관된 하위 섹션 링크는 페이지 상단에 위치해

있습니다. 가상 서버와 관련된 하위 섹션을 설정하기 전에 이 페이지를 작성하고 **ACCEPT** 버튼을 클릭합니다.

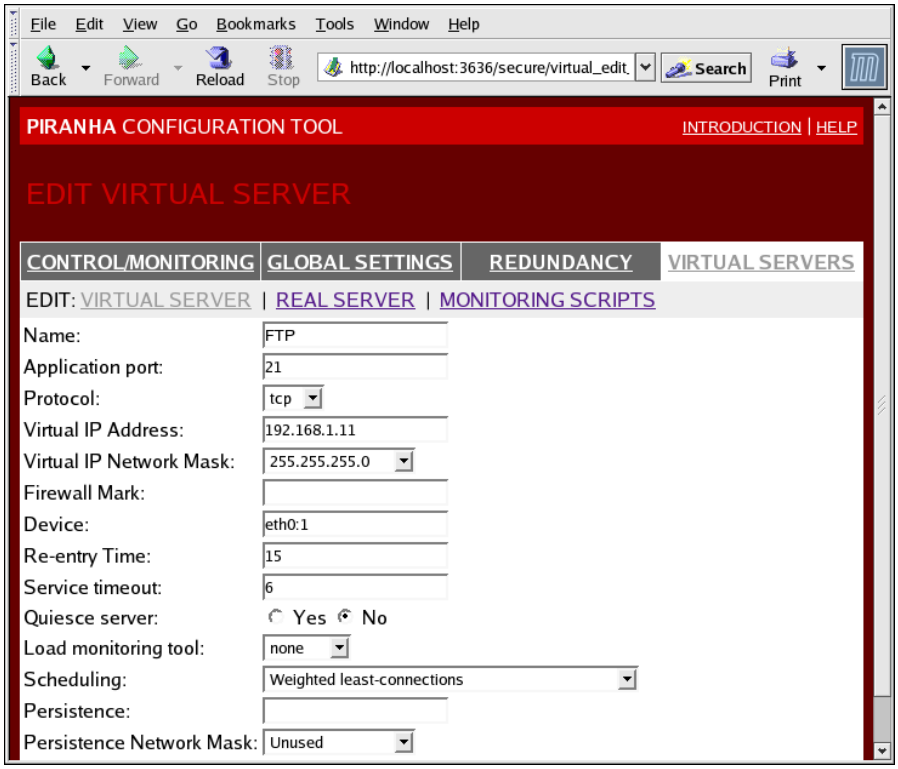

그림 4.6. VIRTUAL SERVER 하위 섹션

NAME

가상 서버를 확인하기 위한 풀어쓴 이름이어야 합니다. 이 이름은 해당 컴퓨터의 호스트명이 *아니므로*, 더 설명적이고 식별하기 쉬운 것으로 정해야 합니다. HTTP와 같이 가상 서버로 사용된 프로토콜을 참조할 수도 있습니다.

Application port

서비스 어플리케이션이 청취할 포트 번호를 입력합니다. 예에서는 HTTP 서비스에 대해 포트 80이 사용되고 있습니다.

Protocol

드롭 다운 메뉴에서 UDP 또는 TCP를 선택합니다. 일반적으로 웹 서버는 TCP 프로토콜이 사용되므로 위의 예에서는 TCP가 선택되어 있습니다.

Virtual IP Address

가상 서버의 유동 IP 주소를 입력합니다.

Virtual IP Network Mask

드롭 다운 메뉴에 있는 가상 서버의 넷 마스크를 설정합니다.

Firewall Mark

다중 포트 프로토콜을 묶거나 프로토콜과 관련하여 개별적으로 다중 포트 가상 서버를 생성하는 것이 아니라면, 이 란에 방화벽 표지 기능 정수값을 입력하지 않습니다. 예에서는 실제로 80이라는 방

화벽 표지 기능값을 사용하여 포트 80 HTTP 연결과 포트 443 HTTPS 연결을 묶고 있기 때문에, 80이라는 **Firewall Mark**를 갖습니다. 지속성 기능과 결합하여 사용할 때, 이 기능은 사용자가 보안이 적용된 웹 페이지나 보안이 적용되지 않은 페이지로 액세스하는 경우 모두 동일한 실제 서버로 라우트되어 상태를 유지할 수 있도록 보장해 줍니다.

경고

이 란에 방화벽 표지 기능을 입력하여 IPVS가 이 방화벽 표지를 사용하는 패킷을 동일하게 취급하도록 하지만, 방화벽 표지 기능을 지정하기 위해서는 **Piranha Configuration Tool** 외부에서도 더 자세한 설정을 해야 합니다. 다중 포트 서비스를 생성하는 방법에 대한 내용은 3.4절. “다중 포트 서비스 및 로드 밸런서 추가 기능 ”에서 참조하시고 고가용성 FTP 가상 서버를 생성하는 방법에 대한 내용은 3.5절. “FTP 설정 ”에서 참조하시기 바랍니다.

Device

Virtual IP Address란에 지정된 유동 IP주소에 바인드할 네트워크 장치 이름을 입력합니다.

공개 네트워크에 연결된 이더넷 인터페이스로 공개적 유동 IP 주소를 알리아스해야만 합니다. 예를 들어, 공개 네트워크가 **eth0** 인터페이스에 있을 경우, **eth0:1**을 장치 이름으로 입력해야 합니다.

Re-entry Time

실제 서버 연결에 실패한 후, 활성 LVS 라우터가 풀로 (실패를 일으켰던) 실제 서버를 다시 불러오기 전 까지의 시간(단위:초)을 정수 값으로 입력합니다.

Service Timeout

실제 서버가 풀에서 종료되었거나 삭제되었다고 판단하기 까지의 시간 (단위:초)을 정수 값으로 입력합니다.

Quiesce server

Quiesce server 라디오 버튼이 선택되었을 경우, 아무때나 새로운 실제 서버 노드가 온라인이 될 수 있으며, 최소 연결 테이블은 0으로 재설정되어 활성 LVS 라우터는 모든 실제 서버가 새로 풀에 추가된 것처럼 요청을 보냅니다. 이러한 옵션은 새로 추가되는 서버가 최초로 풀에 들어왔을 때 접속이 많아져서 과부하에 걸리지 않도록 해 줍니다.

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.

경고

로드 모니터링은 로드 밸런싱과 동일하지 *않으며* 가중치 기반 스케줄링 알고리즘과 결합하려 할 때 스케줄링 상태를 예상하기 어렵게 할 수 있습니다. 또한 로드 모니터링을 사용할 경우, 실제 서버는 반드시 Linux를 사용하는 컴퓨터여야 합니다.

Scheduling

드롭 다운 메뉴에서 선호하는 스케줄링 알고리즘을 선택합니다. 기본값은 **Weighted least-connection**으로 되어 있습니다. 스케줄링 알고리즘에 대한 보다 자세한 내용은 1.3.1절. “스케줄링 알고리즘 ”에서 참조하시기 바랍니다.

Persistence

클라이언트 트랜잭션 동안 관리자가 가상 서버에 지속적으로 연결해야 할 경우, 연결 타임 아웃을 판정하기 까지 허용된 비활성화 시간(단위:초)을 지정합니다.

중요

Firewall Mark 란에 값을 입력할 경우, 지속성 기능에 대한 값도 입력해야 합니다. 또한, 방화벽 표지 기능과 지속성 기능을 함께 사용할 경우, 지속성 기능의 양이 각각의 가상 서버에 대해 방화벽 표지 기능의 양과 같아야 합니다. 지속성 기능 및 방화벽 표지 기능에 대한 보다 자세한 내용은 1.5절. “지속성 및 방화벽 표시 기능 ”에서 참조하시기 바랍니다.

Persistence Network Mask

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

알림

방화벽 표지 기능이 생겨나기 전에는, 서브넷으로 지속성을 제한하는 것이 연결을 묶는 원시적인 방법이었습니다. 이제는, 방화벽 표지와 지속성을 관련시켜서 사용하는 것이 동일한 결과를 얻는 가장 좋은 방법입니다.

경고

새 패널을 선택했을 때 이 패널에서 변경한 사항을 잃지 않으려면 **ACCEPT** 버튼을 클릭해야 합니다.

4.6.2. REAL SERVER 하위 섹션

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

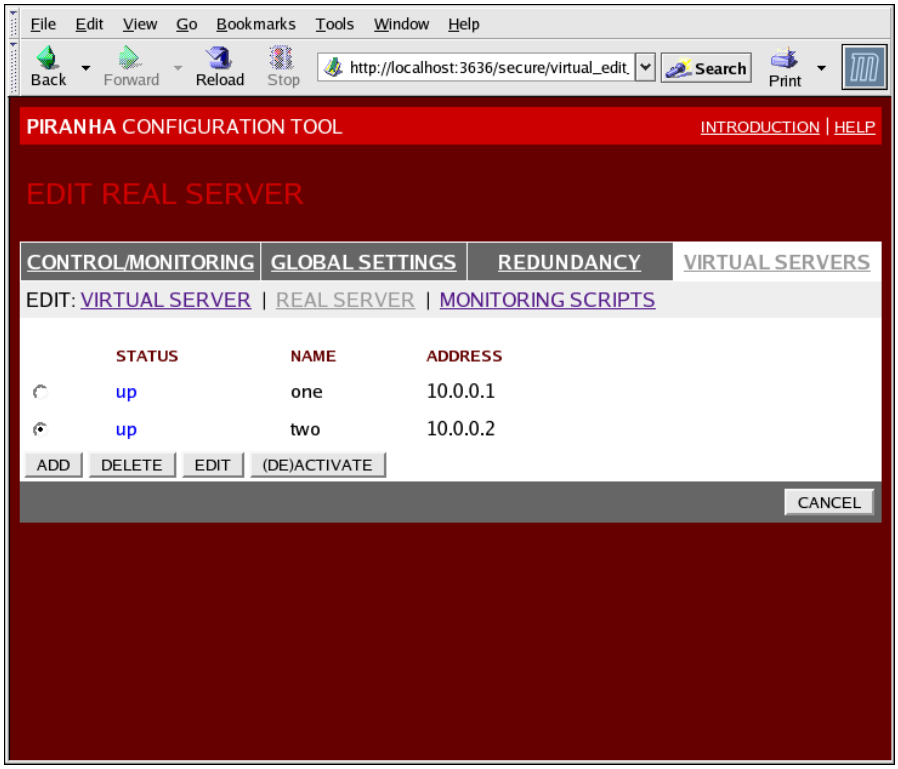

그림 4.7. REAL SERVER 하위 섹션

ADD 버튼을 클릭하여 새 서버를 추가합니다. 기존 서버를 삭제하려면 라디오 버튼을 선택하고 **DELETE** 버튼을 클릭합니다. 그림 4.8. "**REAL SERVER** 설정 패널"에서 볼 수 있듯이 **EDIT** 버튼을 클릭하여 **EDIT REAL SERVER** 패널을 읽어옵니다.

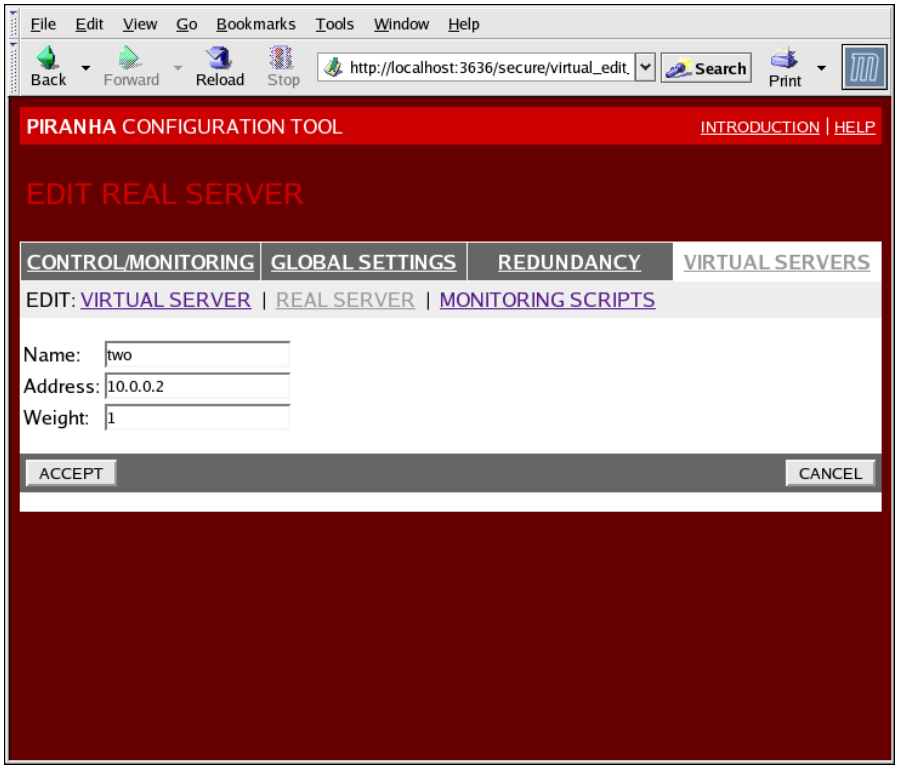

그림 4.8. REAL SERVER 설정 패널

이 패널은 세 가지 항목으로 구성되어 있습니다:

NAME

실제 서버에 해당하는 설명적인 이름입니다.

알림

이 이름은 컴퓨터의 호스트명이 *아니므로* 설명적이고 쉽게 식별할 수 있는 것이어야 합니다.

Address

실제 서버의 IP 주소입니다. 연관된 가상 서버에 대해 청취 중인 포트가 이미 지정되어 있으면, 포트 번호를 추가하지 않습니다.

Weight

풀에 있는 다른 호스트에 대한 상대적인 호스트 용량을 나타내는 정수값입니다. 임의의 값을 지정할 수 있으나 풀에 있는 다른 실제 서버에 대한 비율로 다루어져야 합니다. 서버 가중치에 대한 자세한 내용은 1.3.2절. “서버 가중치 및 스케줄링 ”에서 참조하시기 바랍니다.

경고

새 패널을 선택했을 때 이 패널에서 변경한 사항을 잃지 않으려면 **ACCEPT** 버튼을 클릭해야 합니다.

4.6.3. EDIT MONITORING SCRIPTS 하위 섹션

페이지 상단의 **MONITORING SCRIPTS** 링크를 클릭합니다. **EDIT MONITORING SCRIPTS** 하위 섹션에서 관리자는 가상 서버의 서비스가 각각의 실제 서버에서 잘 작동하는 지를 확인하기 위해 send/expect 문자열 순서를 정할 수 있습니다. 또한 여기서 데이터가 동적으로 변하는 서비스를 확인하기 위한 사용자 정의 스크립트를 지정할 수도 있습니다.

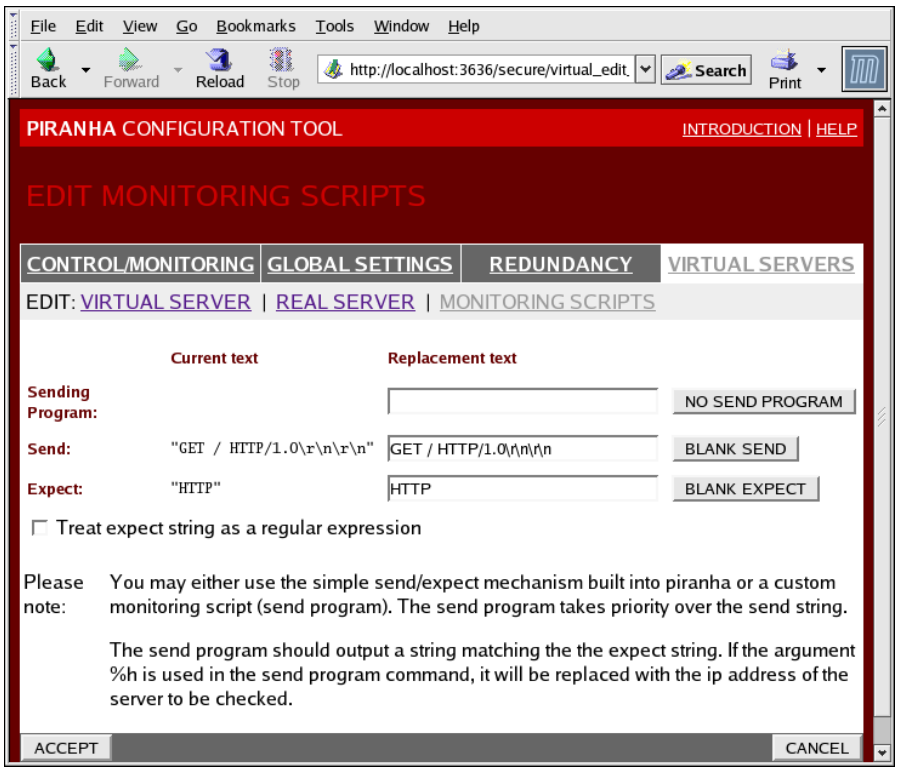

그림 4.9. **EDIT MONITORING SCRIPTS** 하위 섹션

Sending Program

보다 고급 서비스 검증을 위해, 이 란에 서비스 확인 스크립트에 대한 경로를 지정할 수 있습니다. 특히 이러한 기능은 HTTPS나 SSL과 같이 데이터가 동적으로 변화하는 서비스에 유용합니다.

이 기능을 사용하려면 텍스트로 된 응답을 출력하도록 스크립트를 작성하고, 실행 가능하도록 설정한 다음에, **Sending Program** 란에 스크립트의 경로를 입력해야 합니다.

알림

실제 서버 풀에 있는 각각의 서버가 검증되었는 지를 살펴보려면 **Sending Program** 란의 스크립트 경로 뒤에 특수 토큰 **%h**를 사용합니다. 이 토큰은 스크립트가 **nanny** 데몬에 의해 호출될 때, 각각의 실제 서버의 IP주소로 대체됩니다.

다음은 외부 서비스 확인 스크립트 작성 시 가이드로 사용할 수 있는 예시 스크립트입니다:

```
#!/bin/sh

TEST=`dig -t soa example.com @$1 | grep -c dns.example.com

if [ $TEST != "1" ]; then
 echo "OK
else
 echo "FAIL"
fi
```

알림

외부 프로그램이 **Sending Program** 란에 입력되어 있을 경우 **Send** 란은 무시됩니다.

Send

이 란에는 **nanny** 데몬이 각각의 실제 서버로 보낼 문자열을 입력합니다. 기본값으로 이 란은 HTTP에 대해 작성되어 있습니다. 여러분은 필요에 따라 이를 변경할 수 있습니다. 이 란을 빈 칸으로 비워두었을 경우, **nanny** 데몬은 서비스의 포트가 열리는 경우 그 서비스가 실행중이라고 판정하게 됩니다.

이 란에는 하나의 전송 순서만이 허용되며, 인쇄할 수 있는 ASCII 문자와 다음과 같은 이스케이프 문자열만 사용 가능합니다:

- \n은 개행문자
- \r은 캐리지 리턴
- \t은 탭
- \은 바로 다음에 오는 문자를 이스케이프함

Expect

서버가 올바르게 작동할 경우 서버가 리턴할 텍스트 응답을 입력합니다. 만약 독자적인 전송 프로그램을 사용할 경우, 전송 명령이 성공적으로 실행되었을 경우 그 명령이 돌려줄 응답을 입력합니다.

알림

서비스를 호출하기 위해 무엇을 사용할 지를 결정하기 위해, 실제 서버에서 포트로 **telnet** 연결을 오픈한 후 리턴값을 확인할 수 있습니다. 예를 들어, FTP가 220을 들어오는 연결에 대해 보고한다면 **Send**란에 **quit**를 입력하고 **Expect** 란에 **220**을 입력합니다.

경고

새 패널을 선택했을 때 이 패널에서 변경한 사항을 잃지 않으려면 **ACCEPT** 버튼을 클릭해야 합니다.

Piranha Configuration Tool를 사용하여 가상 서버를 설정한 후, 백업 LVS 라우터에 특정 설정 파일을 복사해 두어야 합니다. 보다 자세한 내용은 4.7절. “설정 파일 동기화”에서 참조하시기 바랍니다.

4.7. 설정 파일 동기화

주 LVS 라우터를 설정한 후 로드 밸런서 추가 기능을 시작하기 전에 백업 LVS 라우터에 몇몇 설정 파일을 복사해야만 합니다.

이러한 파일에는 다음과 같은 것이 있습니다:

- **/etc/sysconfig/ha/lvs.cf** — LVS 라우터 용 설정 파일

- **/etc/sysctl** — 커널에서 패킷 포워딩을 활성화시키는 설정 파일
- **/etc/sysconfig/iptables** — 방화벽 표지 기능을 사용할 경우, 사용 중인 네트워크 패킷 필터에 따라서 이 파일 중 하나를 동기화시켜야 합니다.

중요

Piranha Configuration Tool를 사용하여 로드 밸런서 추가 기능을 설정할 때 **/etc/sysctl.conf** 및 **/etc/sysconfig/iptables** 파일은 변경하지 *않습니다.*

4.7.1. lvs.cf 동기화

매번 LVS 설정 파일, **/etc/sysconfig/ha/lvs.cf**을 생성하거나 업데이트할 때 마다 백업 LVS 라우터 노드에 이 파일을 복사해 두어야 합니다.

경고

활성화된 LVS 라우터 노드와 백업 LVS 라우터 노드에는 동일한 **lvs.cf** 파일이 있어야 합니다. LVS 라우터 노드들 사이에 LVS 설정 파일이 일치하지 않을 경우 장애조치가 실행되지 않을 수 있습니다.

설정 파일을 일치시키기 위한 가장 좋은 방법은 **scp** 명령을 사용하는 것입니다.

중요

scp를 사용하려면 **sshd**가 백업 라우터에서 실행되고 있어야 합니다. LVS 라우터에 필요한 서비스를 올바르게 설정하는 방법에 관한 보다 자세한 내용은 2.1절. “LVS 라우터 상의 설정 서비스 ”에서 참조하시기 바랍니다.

라우터 노드에서 **lvs.cf** 파일을 동기화하기 위해 주 LVS 라우터에서 root 사용자로 다음과 같은 명령을 실행합니다:

```
scp /etc/sysconfig/ha/lvs.cf n.n.n.n:/etc/sysconfig/ha/lvs.cf
```

명령에서 ***n.n.n.n***을 백업 LVS 라우터의 실제 IP 주소로 대체합니다.

4.7.2. sysctl 동기화

대부분의 경우 **sysctl** 파일은 한 번만 수정됩니다. 이 파일은 부팅 시 읽어들여지며, 커널에 패킷 포워딩을 활성화하도록 알려줍니다.

중요

커널에서 패킷 포워딩이 활성화 되었는지 잘 모르는 경우, 2.5절. “패킷 포워딩 활성화 ”에서 활성화 여부 확인 방법과 이 중요한 기능을 활성화하는 방법에 대한 지시 사항을 참조하시기 바랍니다.

4.7.3. 네트워크 패킷 필터링 규칙 동기화

iptables을 사용하고 있을 경우, 백업 LVS 라우터에 올바를 설정 파일을 동기화해야 합니다.

네트워크 필터 규칙을 변경했을 경우, 주 LVS 라우터에서 root로 다음과 같은 명령을 입력합니다:

```
scp /etc/sysconfig/iptables n.n.n.n:/etc/sysconfig/
```

명령에서 ***n.n.n.n***을 백업 LVS 라우터의 실제 IP 주소로 대체합니다.

그 다음, 백업 라우터에 **ssh** 세션을 열거나, 또는 root로 컴퓨터에 로그인한 후 다음과 같은 명령을 입력합니다:

```
/sbin/service iptables restart
```

일단 이러한 파일을 백업 라우터에 복사하고, 필요한 서비스들을 시작하셨다면(자세한 내용은 2.1절. “LVS 라우터 상의 설정 서비스”에서 참조), 로드 밸런서 추가 기능을 시작할 준비가 된 것입니다.

4.8. 로드 밸런서 추가 기능 시작

로드 밸런서 추가 기능을 시작하려면, 주 LVS 라우터에 두 개의 root 터미널을 동시에 열거나, 주 LVS 라우터로 **ssh** 세션을 동시에 두 개 여는 것이 가장 좋습니다.

하나의 터미널에서, 다음과 같은 명령으로 커널 로그 메세지를 확인합니다:

```
tail -f /var/log/messages
```

그 후 다른 터미널에 다음과 같은 명령을 입력하여 로드 밸런서 추가 기능을 시작합니다:

```
/sbin/service pulse start
```

다음으로 커널 로그 메세지를 표시중인 터미널에서 **pulse** 서비스 시작의 진행 상태를 살펴봅니다. 다음과 같은 메세지가 출력되면 pulse 데몬이 올바르게 시작된 것입니다:

```
gratuitous lvs arps finished
```

/var/log/messages 파일 보기를 정지하려면, **Ctrl+c** 키를 사용합니다.

이제 부터, 주 LVS 라우터는 활성 LVS 라우터로도 됩니다. 이제 로드 밸런서 추가 기능에 요청을 보낼 수 있지만, 로드 밸런서 추가 기능을 서비스하도록 하기 전에 백업 LVS 라우터를 시작해야 합니다. 이를 위해, 이전의 백업 LVS 라우터 노드에서 설명한 절차를 반복하여 실행합니다.

마지막 단계를 완료하면 로드 밸런서 추가 기능 시작되어 실행될 것입니다.

고가용성 추가 기능과 함께 로드 밸런서 추가 기능 사용하기

로드 밸런싱, 데이터 무결성, 어플리케이션 유용성을 제공하는 고가용성 전자상거래 사이트를 사용하기 위해 고가용성 추가 기능과 함께 로드 밸런서 추가 기능을 사용할 수 있습니다.

그림 A.1. "고가용성 추가 기능과 함께 로드 밸런서 추가 기능 사용 "에 있는 설정은 URL을 통해 온라인 상품 주문에 사용된 전자상거래 사이트를 보여주고 있습니다. 클라이언트는 방화벽과 활성화된 LVS 로드 밸런싱 라우터를 통해 URL에 요청을 보내고, 이 요청은 웹 서버 중 하나로 전달됩니다. 고가용성 추가 기능 노드는 웹 서버로 동적인 데이터를 제공하며, 웹서버는 요청한 클라이언트에게 그 데이터를 전달합니다.

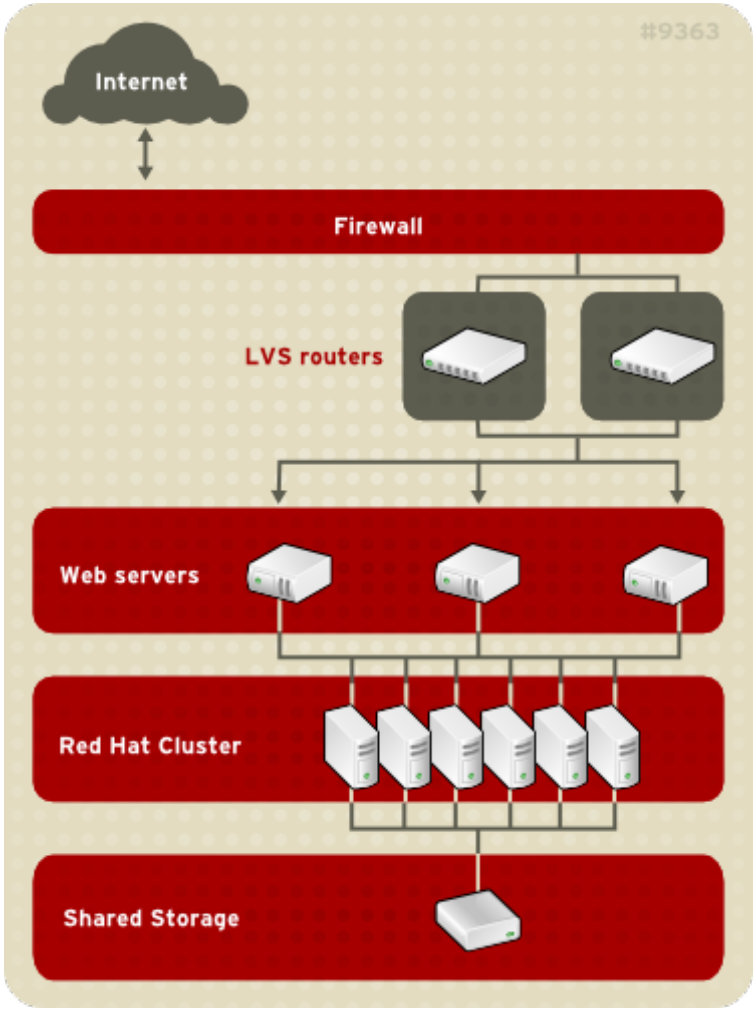

그림 A.1. 고가용성 추가 기능과 함께 로드 밸런서 추가 기능 사용

로드 밸런서 추가 기능으로 동적인 웹 콘텐트를 사용하기 위해서는 (그림 A.1. "고가용성 추가 기능과 함께 로드 밸런서 추가 기능 사용 "에서 보여주듯이) 3-tier 설정이 필요합니다. 이러한 로드 밸런서 추가 기능과 고가용성 추가 기능의 조합은 단일 오류 지점이 없는 높은 통합성의 전자 상거래 사이트를 설정할 수 있도록 합니다. 고가용성 추가 기능을 활용해, 웹 서버에서 네트워크 액세스 가능한 데이터 베이스나 데이터 베이스의 모음의 고가용성을 달성할 수 있습니다.

동적인 컨텐트를 제공하기 위해서는 3-tier 설정이 필요합니다. 웹 서버가 (데이터 변경이 적은) 정적 웹 컨텐트를 서비스할 경우 2-tier 로드 밸런서 추가 기능 설정이 적합한 반면, 동적인 컨텐트를 서비스할 경우에는 2-tier 설정이 적합하지 않습니다. 동적인 컨텐트에는 제품 재고, 구매 주문, 또는 고객 용 데이터베이스등이 포함될 수 있으며, 고객이 최신의 정확한 정보를 액세스를 할 수 있도록 하기 위해, 이 정보는 모든 웹 서버에서 일치해야 합니다.

각각의 tier는 다음과 같은 기능을 제공합니다:

- 첫번째 tier — 웹 요청을 배분하기 위해 로드 밸런싱을 실행하는 LVS 라우터
- 두번째 tier — 요청을 실행하는 웹 서버 모음
- 세번째 tier — 웹 서버로 데이터를 보내는 고가용성 추가 기능

그림 A.1. “고가용성 추가 기능과 함께 로드 밸런서 추가 기능 사용 ”와 같은 로드 밸런서 추가 기능 설정에서 클라이언트 시스템은 월드 와이드 웹 (World Wide Web)상에서 요청을 발생시킵니다. 보안상의 이유로, 이러한 요청은 방화벽을 통과하여 웹 사이트로 들어가는데, 이 방화벽은 전용 방화벽이거나 해당 기능을 제공하는 Linux 시스템이 될 수 있습니다. 이중으로 장애처리(failover)를 하도록 방화벽을 설정할 수도 있습니다. 방화벽 뒤에는 활성 대기(active-standby) 모드로 설정될 수 있는 LVS 로드 밸런싱 라우터가 있습니다. 활성화된 로드 밸런싱 라우터는 웹 서버 집합으로 요청을 전달합니다.

각각의 웹 서버는 개별적으로 클라이언트로부터의 HTTP 요청을 처리하고, 응답을 보낼 수 있습니다. 로드 밸런서 추가 기능은 LVS 라우터 뒤에 웹 서버를 추가하여 웹 사이트의 용량을 확장할 수 있도록 합니다; LVS 라우터는 웹 서버 셋 전반에 걸쳐 로드 밸런싱을 실행합니다. 또한, 어떤 웹 서버에 오류가 난 경우,그 서버는 제거될 수 있습니다; 이 경우 로드 밸런서 추가 기능은 더 작아진 웹 서버 집합 전체에 걸쳐 로드 밸런싱을 계속 하게 됩니다.

개정 이력

고침 6-3.400	**2013-10-31**	**Rüdiger Landmann**
Rebuild with publican 4.0.0		
고침 6-3	**2012-07-18**	**Anthony Towns**
Rebuild for Publican 3.0		
고침 1.0-0	**Wed Nov 10 2010**	**Paul Kennedy**
Red Hat Enterprise Linux 6 버전 최초 릴리즈		

색인

Symbols

N

P

S