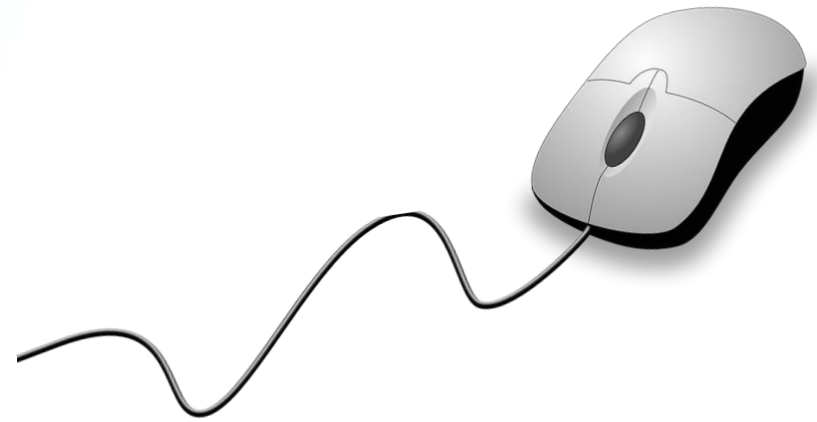


# 공개SW 솔루션 설치 & 활용 가이드

미들웨어 > 분산시스템SW



## 제대로 배워보자

How to Use Open Source Software

---

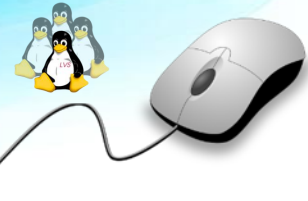
Open Source Software Installation & Application Guide



# CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

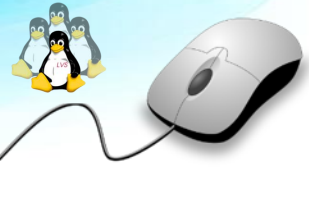
# 1. 개요



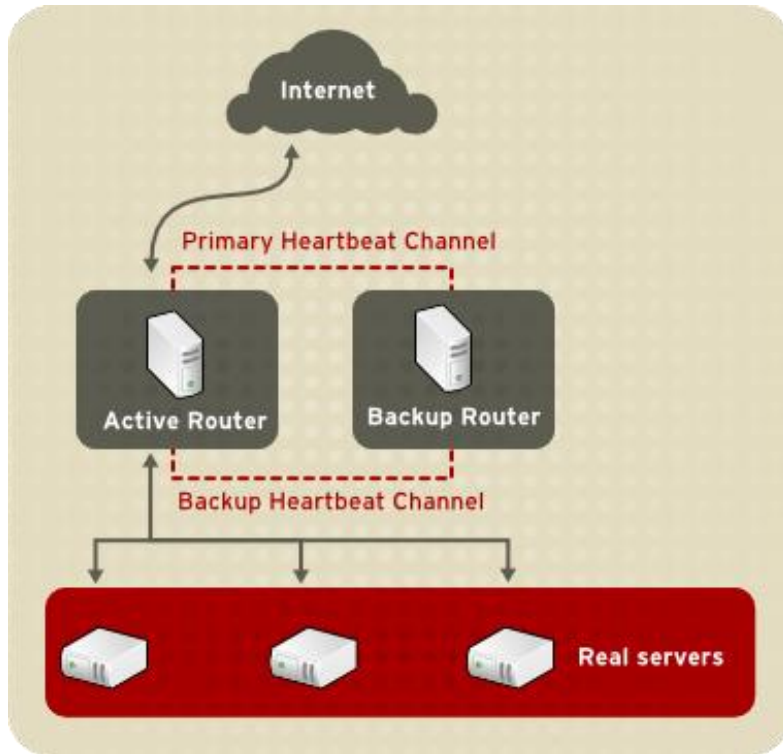
<p><b>소개</b></p>	<ul style="list-style-type: none"> <li>• LVS(Linux Virtual Server)란 한대의 서버로 증가하는 인터넷 사용자를 처리하기가 힘들어 지면서 고 가용성 서버를 구축하기 위해 리눅스 머신을 로드 밸런스 하도록 해주는 운영 시스템</li> <li>• 만약 하나의 노드에서 처리량이 너무 많아서 서비스가 불가능 할 경우 간단히 하나의 노드를 병렬구성으로 추가함으로써 부하분산을 하도록 하는 것을 말함</li> </ul>		
<p><b>주요기능</b></p>	<ul style="list-style-type: none"> <li>• 우수한 확장성, 신뢰성 및 서비스 가능성을 제공하는 클러스터링 기술을 사용하여 Linux용 고성능 및 가용성이 뛰어난 서버를 구축</li> </ul>		
<p><b>대분류</b></p>	<ul style="list-style-type: none"> <li>• 미들웨어</li> </ul>	<p><b>소분류</b></p>	<ul style="list-style-type: none"> <li>• 분산시스템SW</li> </ul>
<p><b>라이선스 형태</b></p>	<ul style="list-style-type: none"> <li>• GNU General Public License</li> </ul>	<p><b>사전설치 솔루션</b></p>	<ul style="list-style-type: none"> <li>• N/A</li> </ul>
<p><b>운영체제</b></p>	<ul style="list-style-type: none"> <li>• Linux, Windows, Solaris 등</li> </ul>	<p><b>버전</b></p>	<ul style="list-style-type: none"> <li>• 1.2.1 (2018년 10월 기준)</li> </ul>
<p><b>특징</b></p>	<ul style="list-style-type: none"> <li>• 높은 가용성</li> <li>• 추가 확장성</li> <li>• 비용 효율성</li> <li>• DR, NAT, Tunneling 방식으로 다양한 환경에서 가상서버를 구축 가능</li> </ul>		
<p><b>보안취약점</b></p>	<ul style="list-style-type: none"> <li>• 취약점 ID : CVE-2000-0248</li> <li>• 심각도 : 10.0 HIGH(V2)</li> <li>• 취약점 설명 : passwd.php3 스크립트의 결함과 함께 원격 사용자가 시스템에서 임의의 명령을 실행 가능</li> <li>• 대응방안 : Redhat piranha-gui-0.4.12-1.i386.rpm 패치 및 Redhat Linux 6.2 i386 이상으로 업그레이드</li> <li>• 참고 경로 : <a href="https://www.securityfocus.com/bid/1148/info">https://www.securityfocus.com/bid/1148/info</a></li> </ul>		
<p><b>개발회사/커뮤니티</b></p>	<ul style="list-style-type: none"> <li>• redhat</li> </ul>		
<p><b>공식 홈페이지</b></p>	<ul style="list-style-type: none"> <li>• <a href="http://www.linuxvirtualserver.org">http://www.linuxvirtualserver.org</a></li> </ul>		



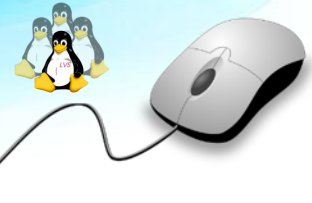
## 2. 기능요약



- 기본 LVS 구성
- LVS 스케줄링 알고리즘
- 라우팅 방식
- 지속성 및 방화벽 표시 기능



# 3. 실행환경



- LVS 지원 운영체제

구분	지원여부
Windows	지원
Solaris	지원
Red Hat	지원
CentOS	지원
Ubuntu	지원
devian	지원
Fedora	지원

# 4. 설치 및 실행

## 세부 목차



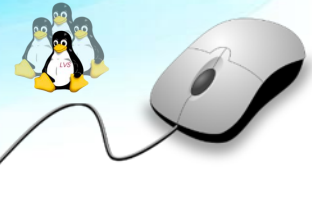
4.1 LVS 설치

4.2 ipvsadm 이란?

4.3 ipvsadm 설치 및 설정



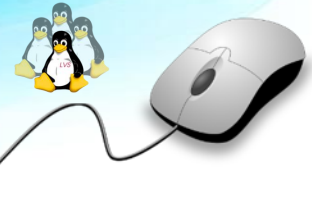
# 4. 설치 및 실행



## 4.1 LVS 설치

- 다양한 방법으로 설정하고 실행 할 수 있는데 커맨드로 간단히 실행하는 방법에서부터 설정 파일을 직접 편집해서 실행하는 방법, Piranha를 이용하는 방법, 다른 패키지를 이용해서 설정하고 실행하는 방법, 조합해서 이용하는 방법 등 있음
- 아래의 사이트에 접속하면 다양한 방법들을 확인 가능
  - <http://www.linuxvirtualserver.org/software/index.html>

# 4. 설치 및 실행



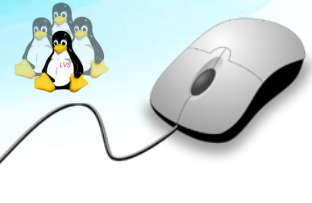
## 4.2 ipvsadm 이란?

- ipvsadm은 LVS(Linux Virtual Server)를 관리하기 위해 사용되는 유틸리티로 iptables가 커널의 netfilter 테이블을 관리하는 것과 유사하게 커널의 Virtual server table을 관리하는데 사용
- ipvsadm은 TCP/UDP를 지원하며 NAT, Tunneling, Direct routing을 지원하고 RR, WRR, LC, WLC를 포함하여 총 8개의 Load balancing 알고리즘 지원





# 4. 설치 및 실행



## 4.3 ipvsadm 설치 및 설정

- ipvsadm 설치 및 설정
  - \$ yum -y install ipvsadm
  - \$ sysctl -a | grep ip\_forward
  - net.ipv4.ip\_forward = 0 으로 나온다면 1 로 고쳐줌
  
  - \$ vi /etc/sysctl.conf
  - net.ipv4.ip\_forward = 1
  
  - \$ sysctl -p
  
- 리부트 후 값이 저장되었는지 확인 진행
  - \$ reboot
  - \$ sysctl -a | grep ip\_forward

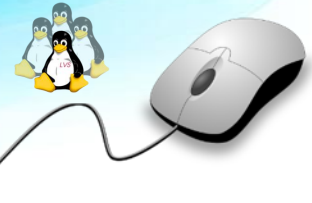
# 5. 기능소개

## 세부 목차



- 5.1 스케줄링이 필요한 이유
- 5.2 LVS 스케줄링
- 5.3 라우팅이 필요한 이유
- 5.4 라우팅 방식
- 5.5 지속성 및 방화벽 표시 기능

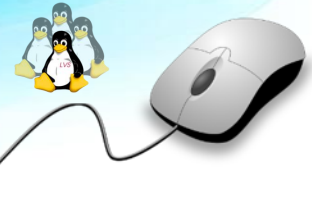
# 5. 기능소개



## 5.1 스케줄링이 필요한 이유

- 실서버를 분산처리 할 때 동일하게 분산하면 사양이 서로 다른 서버가 혼재하는 환경에서는 한쪽으로 부하가 몰릴 수 있음
- 따라서 관리자는 실제 서버 풀에 있는 각각의 노드에 가중치를 부여 가능
- 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 로드 함
  
- 가중치는 서로에 관해 비율로 동작함
  - 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 연결됨
  - 실제 서버 가중치에 해당하는 기본 값은 1임

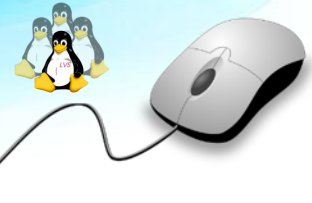
# 5. 기능소개



## 5.2 LVS 스케줄링(1/3)

- Round-Robin Scheduling
  - 각각의 요청을 실제 서버의 풀로 순차적으로 배분하는 방식
  - Round-Robin 알고리즘을 사용하여 용량 이나 부하량을 고려하지 않고 모든 실제 서버를 동일하게 다룸
  - 이러한 스케줄링 모델은 Round-Robin DNS 방식과 유사하지만 호스트 기반이 아닌 네트워크 연결 기반의 보다 세분화된 방식
  - LVS 라운드 로빈 스케줄링은 캐쉬된 DNS 쿼리에 의해 불균등하게 분산되지 않음
- Weighted Round-Robin Scheduling
  - 가중치 기반 라운드 로빈 스케줄링은 풀에 있는 실제 서버의 용량이 다를 경우 선택할 수 있는 사항이지만, 요청 부하량이 아주 다양할 경우 더 가중치가 있는 서버가 요청 공유량보다 더 많이 응답할 수 있음
- Least-Connection
  - 가장 접속이 적은 실제 서버로 더 많은 요청을 배분하는 방식
  - 동적인 스케줄링 알고리즘 유형 중 하나로 IPVS 테이블을 통해 실제 서버로의 라이브 연결을 추적하기 때문에 요청 부하에 극도로 다양한 처리량이 있을 경우 유용
  - 각각의 노드에 대략 비슷한 처리 용량이 있는 실제 서버 풀의 경우에도 적합

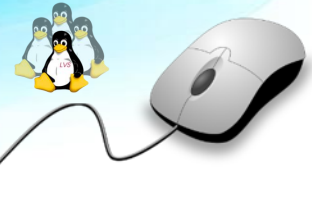
# 5. 기능소개



## 5.2 LVS 스케줄링(2/3)

- Weighted Least-Connections (default)
  - LVS 관리자는 실제 서버 풀에 있는 각각의 노드에 가중치를 부여 가능
  - 가중치는 정수 값으로 모든 가중치 관련 스케줄링 알고리즘 영향을 미치며 LVS 라우터가 다른 처리 용량이 있는 하드웨어를 보다 균등하게 로드
- Locality-Based Least-Connection Scheduling
  - 목적지 IP와 관련하여 가장 접속이 적은 서버로 더 많은 요청을 배분하는 방식
  - 이러한 알고리즘은 프록시-캐쉬 서버 클러스터에서 사용
  - 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 라우팅

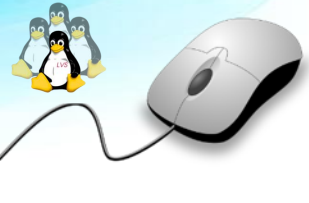
# 5. 기능소개



## 5.2 LVS 스케줄링(3/3)

- Locality-Based Least-Connection Scheduling with Replication Scheduling
  - 목적지 IP와 관련하여 가장 접속이 적은 서버로 더 많은 요청을 배분하는 방식
  - 알고리즘은 프록시-캐쉬 서버 클러스터 사용
  - 대상 IP 주소를 실제 서버 노드의 서브셋으로 묶는 다는 점에서 LBLC 스케줄링과 다름
  - 서브넷에 있는 가장 접속이 적은 서버로 요청을 라우팅
  - 목적지 IP 주소에 대해 모든 노드가 처리 용량을 초과할 경우 가장 접속이 적은 실제 서버를 실제 서버의 전체 풀에서 목적지 IP의 실제 서버에 대한 서브넷으로 추가하여 해당 목적지 IP 주소에 대한 새로운 서버를 복사
  - 최고로 과부화된 노드는 실제 서버 서브넷에서 제외
- Destination Hash Scheduling
  - 정적 해시 테이블에 있는 목적지 IP를 찾아 실제 서버의 풀로 요청을 배분하는 방식
- Source Hash Scheduling
  - 정적 해시 테이블에 있는 소스 IP를 찾아 실제 서버의 풀로 요청을 배분하는 방식

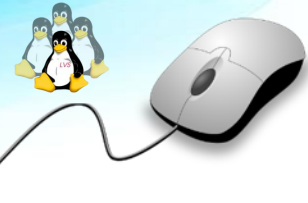
# 5. 기능소개



## 5.3 라우팅이 필요한 이유

- 데이터가 전달하는 과정에서 여러 네트워크들을 통과 해야 하는 경우가 생김
- 여러 네트워크들의 연결을 담당하고 있는 라우터 장비가 데이터의 목적지가 어디인지 확인하여 빠르고 정확한 길을 찾아 전달해주는 것을 라우팅이라 함
  
- 전달받은 패킷을 전달하기 위해 라우터도 가장 먼저 목적지가 어디인지 ip주소를 확인
- 가장 빠른 경로가 어디인지를 확인하고 그 경로로 가기 위해서는 자신이 어느 인터페이스로 패킷을 내보내야 하는지 결정
- 결정한 인터페이스로 패킷을 전달하면 그 패킷은 또 다른 라우터로 전달되어 위와 같은 과정을 목적지 네트워크에 도착할 때까지 반복
- 이처럼 데이터를 목적지까지 전달하기 위한 모든 일련의 과정을 통틀어 라우팅이라 함

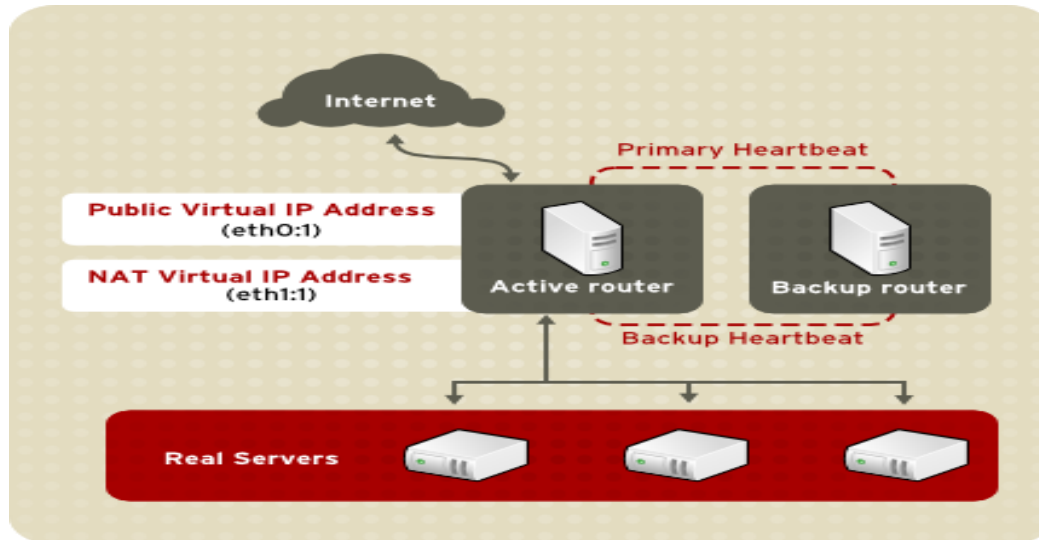
# 5. 기능소개



## 5.4 라우팅 방식(1/4)

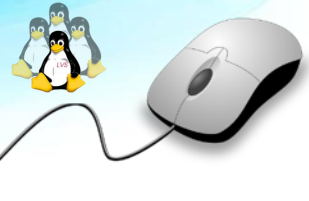
- NAT 라우팅

- 그림에서 활성화된 LVS 라우터에는 두개의 NIC 있음
- 인터넷용 NIC는 eth0에 실제 IP 주소가 있고 알리아스된 eth0:1에는 유동 IP 주소가 있음 개인 네트워크 인터페이스용 NIC에는 eth1에 실제 IP 주소가 있고 알리아스된 eth1:1에는 유동 IP 주소가 있음
- 장애 조치 시 인터넷에 대한 가상 인터페이스와 가상 인터페이스에 대한 개인 네트워크는 동시에 백업 LVS 라우터에 의해 백업
- 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 않음



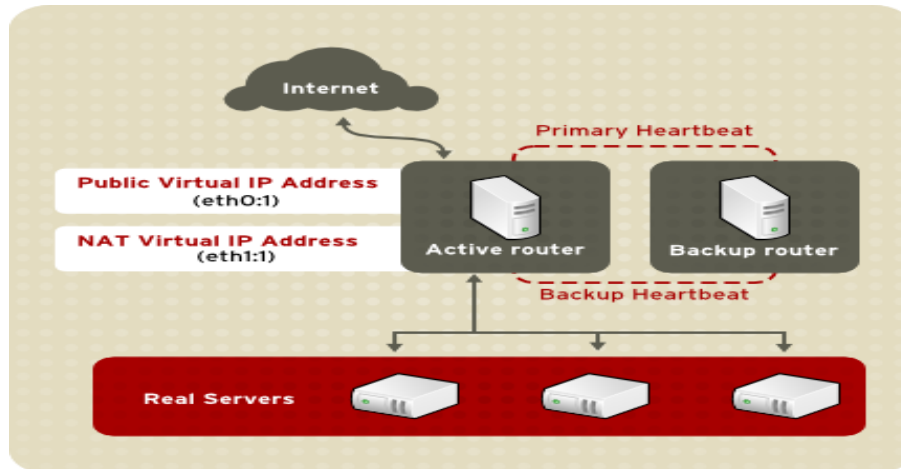


# 5. 기능소개

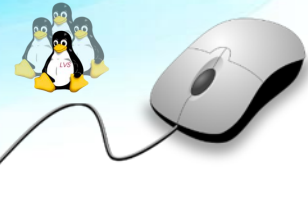


## 5.4 라우팅 방식(2/4)

- 이러한 토폴로지를 사용하여 활성 LVS 라우터는 요청을 받아 적절한 서버로 보냄
  - 그 뒤 실제 서버는 요청을 처리하고 LVS 라우터로 패킷을 되돌려 보냄
  - LVS 라우터는 네트워크 주소 트랜잭션을 사용하여 패킷에 있는 실제 서버의 주소를 LVS 라우터 공개 VIP 주소로 교체
  - 실제 서버의 실질적인 IP 주소가 요청 중인 클라이언트에게 보이지 않게 되므로 이러한 절차를 IP 마스커레이딩이라고 부름
- NAT 라우팅을 사용하면 다양한 운영 체제가 실행되고 있는 아무 컴퓨터나 실제 서버가 될 수 있음
  - NAT 라우팅의 주요 취약점은 들어오고 나가는 모든 요청을 처리해야 하기 때문에 LVS 라우터의 대량의 분배 작업에서 병목 현상이 일어날 수 있다는 점이 있음



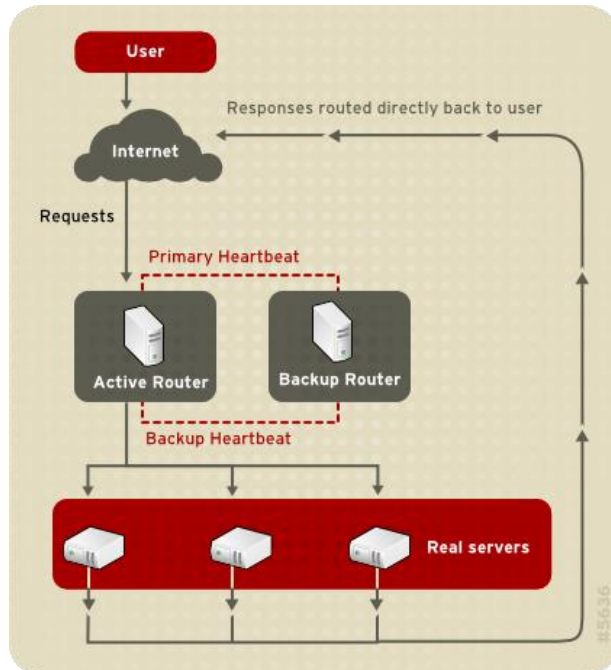
# 5. 기능소개



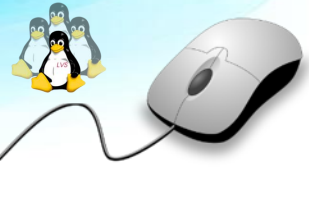
## 5.4 라우팅 방식(3/4)

- 직접 라우팅

- 전형적인 직접 라우팅 LVS 설정에서 LVS 라우터는 가상 IP (VIP)를 통해 들어오는 서버 요청을 받고 스케줄링 알고리즘을 사용하여 실제 서버에 요청을 보냄
- 각각의 실제 서버는 요청을 처리하고 LVS 라우터를 바이패싱하여 클라이언트에게 직접 응답
- 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 있음

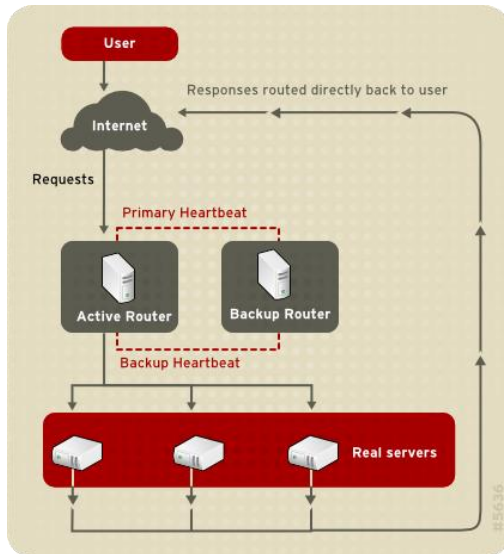


# 5. 기능소개

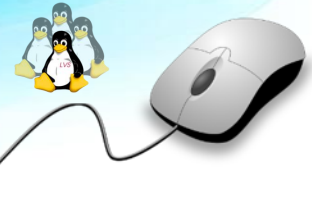


## 5.4 라우팅 방식(4/4)

- 직접 라우팅
  - LVS에서 직접 라우팅을 사용하는데 있어서 여러 장점이 있는 반면 제한점도 있음
  - 직접 라우팅을 통한 LVS 사용에서 가장 일반적으로 발생하는 문제는 ARP 와의 문제임
  - 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 점임
  - LVS 라우터와 실제 서버의 VIP가 같을 경우 ARP 요청은 VIP와 관련되어 있는 모든 컴퓨터로 브로드캐스트됨
  - 이는 VIP가 실제 서버 중 하나에 관련되어 직접적으로 요청을 처리하는 것 LVS 라우터를 바이패싱하여 LVS 설정 목적을 의미 없게 하는 것과 같이 여러가지 문제를 발생시킬 수 있음



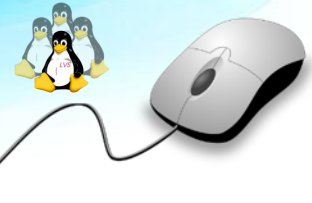
# 5. 기능소개



## 5.5 지속성 및 방화벽 표시 기능(1/2)

- 지속성
  - 활성화된 경우 지속성은 타이머처럼 작동
  - 클라이언트가 서비스에 연결되면 LVS는 지정된 기간 동안의 마지막 연결을 기억
  - 동일한 클라이언트 IP주소가 해당 기간 내에 다시 연결되면 이전에 연결한 서버로 전송되어 로드 밸런싱 메커니즘을 우회
  - 시간 창 밖에서 연결이 발생하면 지정된 예약 규칙에 따라 연결 처리
  - 어떤 주소가 더 높은 수준의 지속성을 갖게 할 지를 제어하기 위한 도구로서 관리자가 클라이언트 IP 주소 테스트를 위한 서브넷 마스크를 지정하게 하여 서브넷에 그룹으로 연결
  - 목적지가 다른 포트에 대한 그룹 연결은 FTP와 같이 하나 이상의 포트를 사용하여 통신하는 프로토콜의 경우 중요한 사항이 될 수 있음
  - 하지만 목적지가 다른 포트로 되어있는 그룹 연결 문제를 처리하는데 지속성 기능이 가장 효과적인 방법은 아님
  - 이러한 경우 방화벽 표시 기능 (firewall marks)을 사용하는 것이 가장 좋음

# 5. 기능소개



## 5.5 지속성 및 방화벽 표시 기능(2/2)

- 방화벽 표시 기능
  - 방화벽 표시 기능은 연관된 프로토콜의 프로토콜이나 그룹에 사용되는 가장 쉽고 효과적인 그룹 포트 방법
  - 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 있음
  - 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 보호 가능
  - 이와 같은 효과적이고 사용 간편함 때문에 LVS 관리자는 그룹 연결을 할 때 마다 지속성 기능보다 방화벽 표시 기능을 사용해야함
  - 하지만 클라이언트가 일정 기간 동안 동일한 가상 서버로 다시 연결하게 하기 위해 가상 서버에 방화벽 기능과 함께 지속성 기능을 추가 가능

# 6. 활용예제

세부 목차

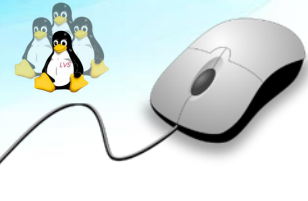


6.1 다이렉트 라우팅 방식

6.2 NAT 라우팅 방식



# 6. 활용예제



## 6.1 다이렉트 라우팅 방식(1/6)

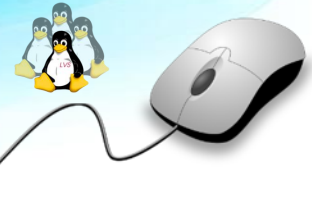
- 가상IP(VIP) 추가
  - \$ ipconfig ens33:0 192.168.0.10 netmask 255.255.255.0 up
- 잘못 추가했다면
  - \$ ip addr del 192.168.0.10 dev ens33:0

- \$ ip addr

```
tuser@localhost:/home/tuser
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
TCP 192.168.0.10:80 wrp persistent 300
-> 192.168.0.101:80      Route 5    0    0
-> 222.117.155.123:80   Route 10   0    5
[root@localhost tuser]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet 192.168.0.10/32 brd 192.168.0.10 scope global lo:0
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:15:2b:98 brd ff:ff:ff:ff:ff:ff
    inet 222.117.155.123/24 brd 222.117.155.255 scope global noprefixroute dynamic ens33
        valid_lft 2506sec preferred_lft 2506sec
    inet 192.168.0.10/24 brd 192.168.0.255 scope global ens33:0
        valid_lft forever preferred_lft forever
    inet6 fe80::9daf:aa9c:a7c6:a2a7/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:51:e1:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:51:e1:91 brd ff:ff:ff:ff:ff:ff
[root@localhost tuser]#
```



# 6. 활용예제

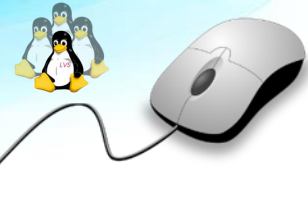


## 6.1 다이렉트 라우팅 방식(2/6)

- Ipv6adm 에 분산할 서비스 등록
  - \$ ipv6adm -C
  - \$ ipv6adm -A -t 192.168.0.10:80 -s wrr -p 300
    - \* -s rr : 라운드로빈
    - \* -s wrr : 가중치 적용 라운드로빈
    - \* -s lc : 최소연결
    - \* -s wlc : 가중치 적용 최소연결
    - \* -p 300 으로 세션 시간을 5분으로 설정
- 이제 분산에 참가 시킬 서버 지정
  - \$ ipv6adm -a -t 192.168.0.10:80 -r 192.168.0.101:80 -g -w 5
  - \$ ipv6adm -a -t 192.168.0.10:80 -r 222.117.155.123:80 -g -w 10



# 6. 활용예제



## 6.1 다이렉트 라우팅 방식(3/6)

- 분산에 참가 시킬 서버 지정

```
- $ ipvsadm -a -t 192.168.0.10:80 -r 192.168.0.101:80 -g -w 5  
- $ ipvsadm -a -t 192.168.0.10:80 -r 222.117.155.123:80 -g -w 10
```

- 설정보기

```
- $ ipvsadm -Ln
```

```
[root@localhost tuser]# ipvsadm -Ln  
IP Virtual Server version 1.2.1 (size=4096)  
Prot LocalAddress:Port Scheduler Flags  
-> RemoteAddress:Port Forward Weight ActiveConn InActConn  
TCP 192.168.0.10:80 wrr persistent 300  
-> 192.168.0.101:80 Route 5 0 0  
-> 222.117.155.123:80 Route 10 0 5
```

- 연결상태보기

```
- $ ipvsadm -Lnc
```

```
[root@localhost tuser]# ipvsadm -Lnc  
IPVS connection entries  
pro expire state source virtual destination
```

- \* -w 로 가중치 줌
- \* -g 다이렉트 라우팅
- \* -i 터널링
- \*-m NAT



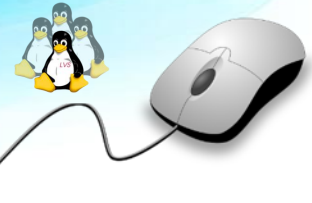
# 6. 활용예제



## 6.1 다이렉트 라우팅 방식(4/6)

- ipvsadm 명령어
  - -A 서비스 추가
  - -E 서비스 수정
  - -D 서비스 삭제
  - -a 리얼서버 추가
  - -e 리얼서버 수정
  - -d 리얼서버 삭제
  - -C 설정 초기화
  - -t 대상 타겟
  - -r 리소스 주소
  - -g 다이렉트 라우팅
  - -w 가중치 설정
  - -m NAT

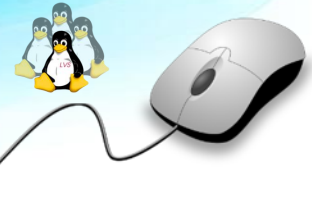
# 6. 활용예제



## 6.1 다이렉트 라우팅 방식(5/6)

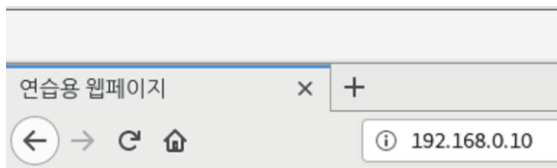
- real 서버 쪽에 VIP 점유를 막기 위한 옵션 설정
  - `$ vi /etc/sysctl.conf`
- ARP 문제 해결을 위해 arp에 대한 응답 없음으로 설정
  - `net.ipv4.conf.lo.arp_ignore = 1`
  - `net.ipv4.conf.lo.arp_announce = 2`
  - `net.ipv4.conf.all.arp_ignore = 1`
  - `net.ipv4.conf.all.arp_announce = 2`
  - `net.ipv4.conf.default.arp_ignore = 1`
  - `net.ipv4.conf.default.arp_announce = 2`
  - `net.ipv4.conf.eth0.arp_ignore = 1`
  - `net.ipv4.conf.eth0.arp_announce = 2`
- `$ sysctl -p`

# 6. 활용예제



## 6.1 다이렉트 라우팅 방식(6/6)

- real 서버쪽에 loop back up 등록
  - \$ touch /etc/sysconfig/network-script/ifcfg-lo:0
- 직접 추가
  - \$ ifconfig lo:0 192.168.0.10 netmask 255.255.255.255 broadcast 192.168.0.10 up
  - \$ route add -host 192.168.0.10 dev lo:0
- 192.168.0.10 접속하면 만들어 두었던 홈페이지 표출 확인



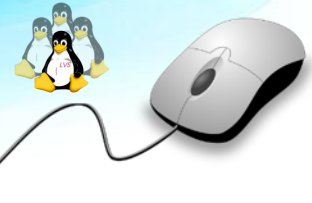
**CentOS 웹서버 구축**

**naom**

**트트**



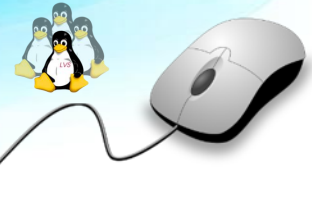
# 6. 활용예제



## 6.2 NAT 라우팅 방식(1/4)

- NAT 방식은 network이 다를 경우나 포트 포워딩이 필요할 경우 사용
- 모든 서버는 eth0 에 사설 대역을 할당해두고, 분산 서버에만 eth1로 리얼망 ip를 추가부여
  - eth1 리얼망 IP (VIP) : 211.47.13.234
- 분산을 위한 가상을 하나 더 올림
  - eth0:0 ip는 10.0.0.234 , 10.0.0.101, 10.0.0.102
- 총 3대의 서버 중 234번은 리얼망과 사설망을 같이 사용하며 분산 서버가 됨

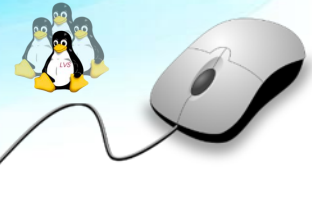
# 6. 활용예제



## 6.2 NAT 라우팅 방식(2/4)

- 가상망 ip 를 셋팅
  - 분산 서버는 ip만 셋팅
    - \$ ifconfig eth0:0 10.0.0.234 netmask 255.255.255.0 up
  - 리얼 서버는 게이트웨이를 분산 서버로 설정
    - \$ ifconfig eth0:0 10.0.0.2 netmask 255.255.255.0 up
    - \$ route add default gw 10.0.0.234 dev eth0:0
  - 다른 게이트웨이가 있으면 오동작을 하게 되므로 제거
    - \$ route del default gw 192.168.0.1

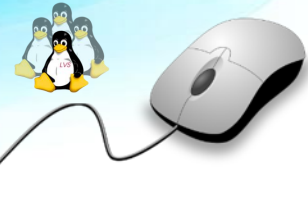
# 6. 활용예제



## 6.2 NAT 라우팅 방식(3/4)

- 가상 서버에 로드 밸런스가 가능하도록 패킷 포워딩 해제
  - \$ vi /etc/sysctl.conf
  - net.ipv4.ip\_forward = 0을 1로 수정
  - \$ sysctl -p
  
- ipvsadm 에 분산할 서비스 등록
  - \$ ipvsadm -C
  - \$ ipvsadm -A -t 211.47.13.234:80 -s wlc

# 6. 활용예제

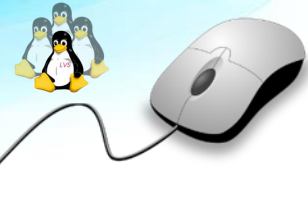


## 6.2 NAT 라우팅 방식(4/4)

- 분산에 참가시킬 서버 지정
  - \$ ipvsadm -a -t 211.47.13.234:80 -r 10.0.0.101:8080 -m -w 1
  - \$ ipvsadm -a -t 211.47.13.234:80 -r 10.0.0.102:8080 -m -w 2
  
  - \$ ipvsadm -Ln
  - \$ ipvsadm -Lnc
- -m을 사용해야 NAT가 됨
- NAT 방식은 gateway 등록 중요
- 2개의 대역을 사용하고 있기 때문에 eth0:0 보다는 가능하다면 eth1 이나 eth2 등을 추가해서 엮어주는 것이 좋음



# 7. FAQ



**Q** LVS 클러스터는 Linux 서버 만 지원합니까?

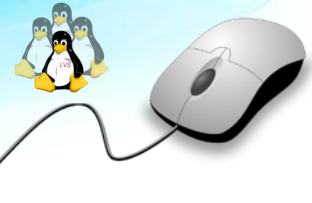
**A** 아닙니다. 실제 서버는 Linux, BSD, Solaris 및 Windows와 같은 LVS 클러스터의 모든 운영 체제를 거의 실행할 수 있습니다. LVS / NAT 는 TCP / IP를 지원하는 운영 체제의 서버를 균형 조정하고 LVS / TUN 은 IP 터널링 프로토콜이 있는 서버를 필요로 하며 LVS / DR 은 arp가 아닌 장치가 있는 서버를 필요로 합니다.

**Q** LVS는 100 만 개가 넘는 동시 연결을 처리 할 수 있습니까?

**A** 예, LVS는 100 만 개가 넘는 동시 연결을 처리 할 수 있습니다. 하나의 연결은 LVS 상자에서 128 바이트를 소비하기 때문에 1G 메모리가 있는 LVS 상자는 800 만 개가 넘는 동시 연결을 처리 할 수 있습니다.



# 8. 용어정리



용어	설명
라운드-로빈 (round-robin)	로드밸런서에 들어오는 요청 패킷들을 차례대로 실제 서버에 할당하는 방식
가중 라운드-로빈 (weighted round-robin)	가중 라운드-로빈 방식은 기본적으로 라운드-로빈 방식을 사용하면서 각 서버에 서로 다른 가중치를 주어서 할당하는 방식
최소 연결 (least connection)	최소 연결 방식은 실제 서버들 중에서 현재 가장 적은 수의 요청을 처리하고 있는 서버를 선택하여 요청 패킷을 할당하는 방식
가중 최소 연결 (weighted least connection)	가중 최소 연결 방식은 기본적으로 최소 연결 방식인데, 가중 라운드-로빈 방식과 마찬가지로 각 서버에 서로 다른 가중치를 주어서 할당하는 방식



# Open Source Software Installation & Application Guide

**nipa** 공개SW역량프라자



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.