
네트워크 분산처리 분야 Stack 통합 Test 결과보고서 [ZenLoadBalancer]

2014. 06.

목 차

I. Stack 통합 테스트 개요	1
1. 목적	1
II.테스트 대상 소개	2
1. ZenLoadBalancer 소개	2
III. Stack 통합 테스트	5
1. 테스트 환경	5
2. 주요 테스트 방법	6
3. 기능 테스트 수행 결과	8
VI. 종합	9
※ 참고자료	9
[별첨1] ZenLoadBalancer 테스트 케이스	

I. Stack 통합 테스트 개요

공개SW Stack 통합테스트는 여러 공개SW들의 조합으로 시스템 Stack을 구성한 후 Stack을 구성하는 공개SW의 상호운용성에 중점을 두고 기능 및 성능테스트 시나리오를 개발하여 테스트를 진행한다.

본 통합테스트를 통해 안정된 Stack 정보를 제공하여 민간 및 공공 정보시스템 도입 시 활용될 수 있도록 한다.

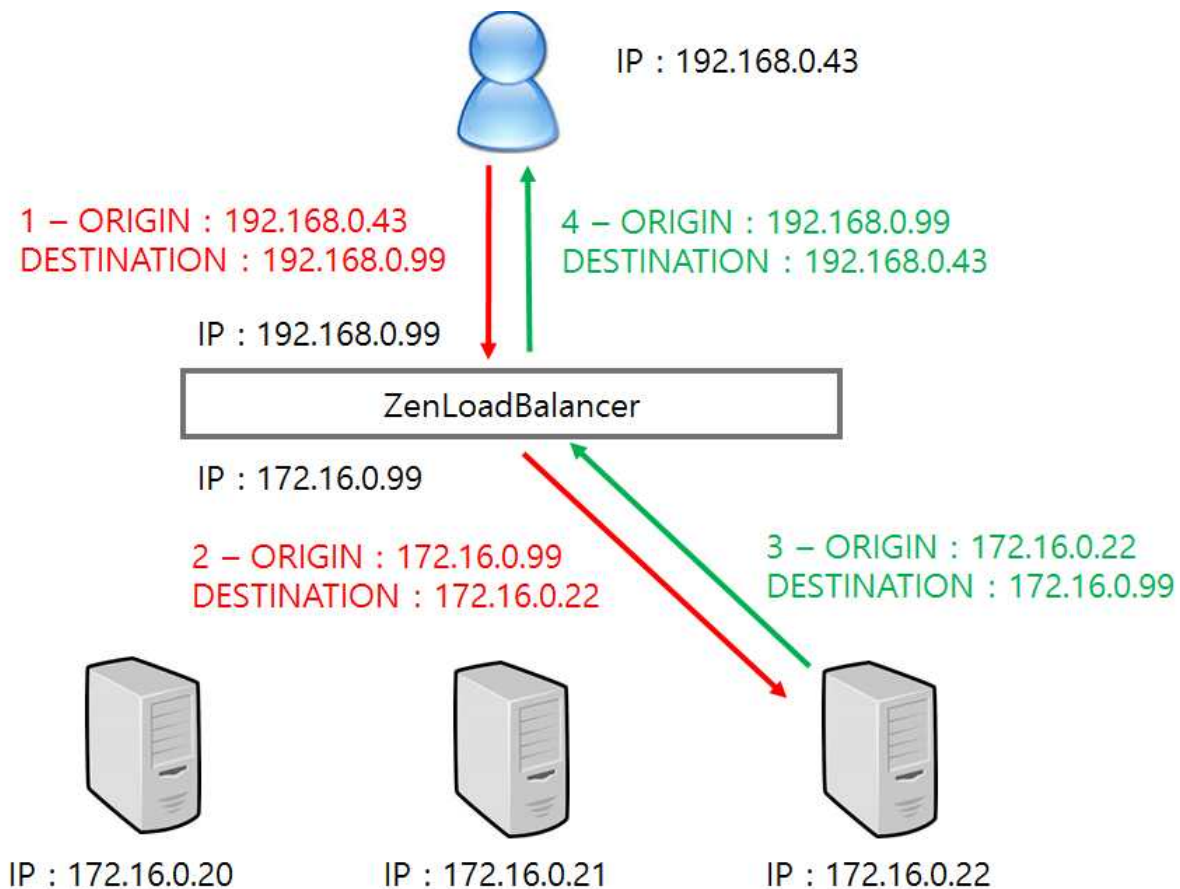
1. 목적

- 공개SW Stack 통합 테스트 수행 목적
 - 공개SW로 구성된 Stack이 유기적으로 잘 동작함을 확인
 - 다양한 Stack 구성에 기반을 둔 테스트를 통해 안정된 Stack 조합 규명
 - 공개SW 시스템 도입을 위한 Stack 참조모델의 신뢰성 정보로 활용
 - 공개SW의 신뢰성과 범용성에 대한 사용자 인식 제고

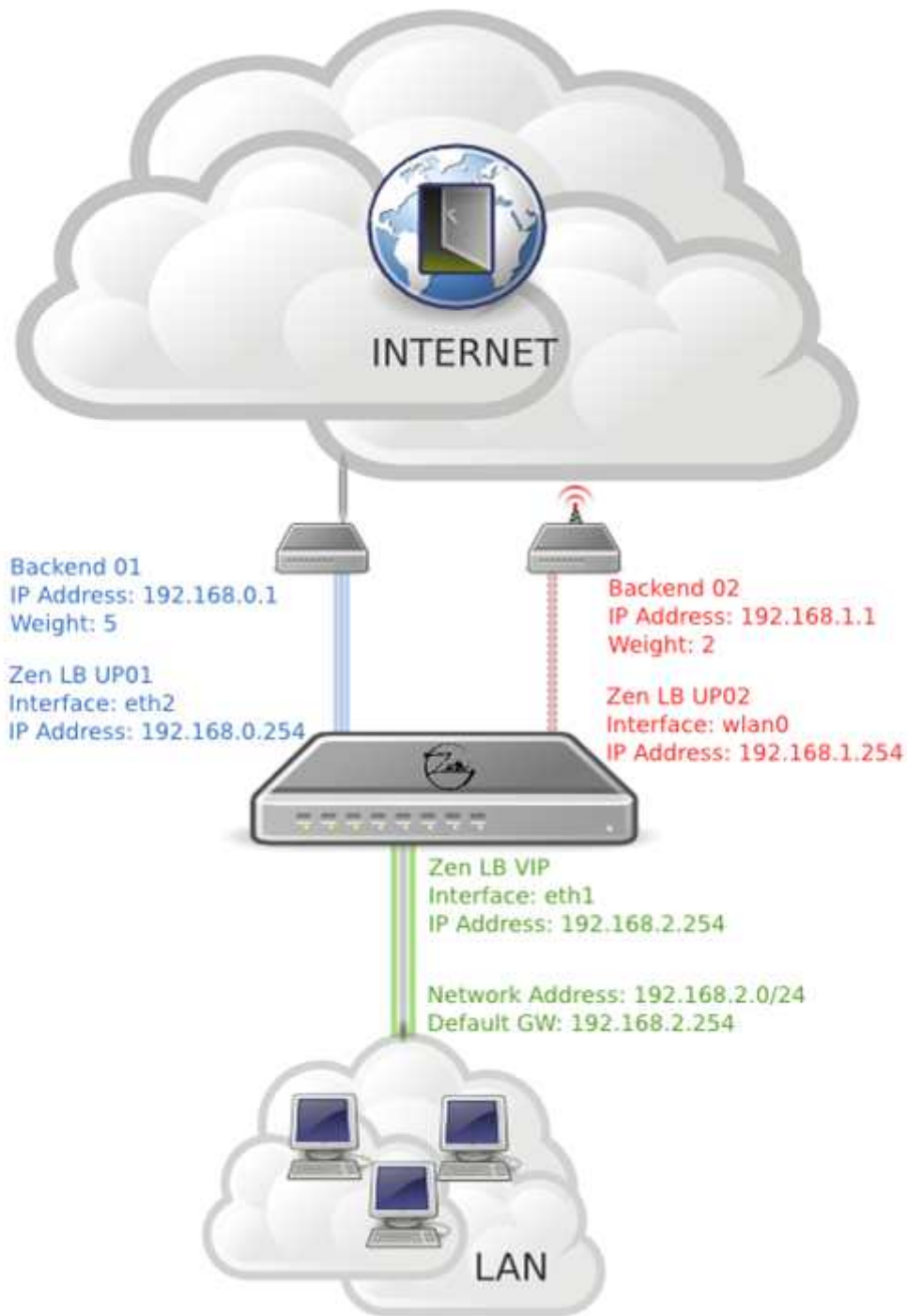
II. 테스트 대상 소개

1. ZenLoadBalancer 소개

ZenLoadBalancer는 오픈소스 기반의 네트워크 분산처리를 지원하는 솔루션이다. ZenLoadBalancer는 소프트웨어 적으로 Farm 설정에 의한 네트워크 그룹 정의 및 서버 정의, 네트워킹(Networking), 클러스터링(Clustering), 모니터링(Monitoring), 보안인증관리(Secure Certificates Management), Log 관리, Configuration 백업, Uplink Load Balancing 등의 다양한 기능을 제공한다.



[그림 1-1. ZenLoadBalancer 기본 환경 구성 - NAT 구조]



[그림 1-2. ZenLoadBalancer 기본 환경 구성 - DATALINK 구조]

※ 상기 환경구성 구조도에 사용된 네트워크 인터페이스 정보는 이해를 돕기 위한 설정으로 본 테스트 진행 상 설정과는 차이가 있음. (그림 1-1 / 그림 1-2 참조)

□ ZenLoadBalancer 기본 개념

- Farm : IP address 및 Port 정보를 포함한 네트워크 인터페이스 정보를 기준으로 정의되며 동일한 서비스를 제공하는 서버를 그룹 관리하는 단위.
- Backend : Farm 정의에 대한 real Service를 제공하는 서버로 Client에 의해 요청되는 모든 real data를 처리함.
- Client : Load Balancer virtual service에 연결되는 Source IP address로 구분되며, 일반적으로 user request 또는 remote process에 의해 초기화됨.
- Real IP : 네트워크 구성에 대해 정의되며, server 또는 NIC에 할당되는 물리적 IP address.
- Virtual IP : Farm 기준으로 정의되며, load balancing node들 간에 전달될 준비가 된 virtual service의 네트워크 구성에 적용되는 유동(floating) IP address.

□ ZenLoadBalancer 시스템 요구사항

항목	정보
지원 OS	- GNU/ Linux
하드웨어 사양	- 최소사양 : N/A - 권장사양 * CPU : 2.0GHz(2Cores) * Memory : 2GB 이상 * HDD : 8GB 이상

[표 1. ZenLoadBalancer 시스템 요구사항]

※ 추가 자세한 정보는 아래의 링크 내역 참조

- <http://freecode.com/projects/zen-load-balancer>
- <http://comments.gmane.org/gmane.comp.web.zenloadbalancer.user/1289>
- <http://www.zenloadbalancer.com/zlb-administration-guide-v304/>

III. Stack 통합 테스트

1. 테스트 환경

ZenLoadBalancer SW 환경

SW	Version
ZenLoadBalancer	3.03

[표 2. 테스트 SW]

※ 참고 - 현재 (2014.07.01기준) zenloadbalancer 공식 사이트(<http://www.zenloadbalancer.com/>) 를 통해 다운로드 및 사용 가능한 버전(v3.03)은 Debian 32bit로 제공되고 있음.

Stack 환경

Stack	OS	IP
A (Server 1)	Linux (Debian 32bit)/ ZenLoadBalancer	10.30.1.100
A (Server 2)	Linux (Debian 32bit)/ ZenLoadBalancer	10.30.1.200
A (Client)	Linux (Ubuntu 12.04LTS 64bit)	121.162.249.28

[표 3. Stack 환경]

※ 참고 - Stack A의 Server1, Server2 설정에 따라 Farm 정의 및 Cluster 구성과 제반 인터페이스 관련 기능 테스트 수행.

HW 환경

제조사	모델명	CPU	MEM	Disk	NIC
HP	DL580G5	Quad-Core 2.4Ghz~4P	8GB	146GB x 5	Gigabit 2Port

[표 4. HW 환경]

※ 참고 - Stack의 각 서버 및 Client 는 동일 사양의 HW를 기반으로 구성되었으며, 네트워크 구성 상 HW 제원 및 방화벽 정책에 따른 제약으로 일부 VM(VirtualBox 4.3.12기준) 설정 적용하여 환경 구성 진행함.

2. 주요 테스트 방법

□ 탐색적 테스트(Exploratory Testing)

탐색적 테스트는 테스트 엔지니어의 지적 능력을 최대한 공유, 활용하는 것을 목적으로 하는 테스트 접근법으로 테스트를 수행할 대상을 실행시켜 사용함과 동시에 사용 측면에서 문제가 되는 부분에 집중하여 테스트를 설계 및 계획한다. 이러한 과정은 효율적 진행을 위한 Time Boxing을 통해 수행되므로 테스트 케이스 작성을 최소화할 수 있고, 상대적으로 적은 시간에 집중적인 테스트를 가능하게 한다.

□ 리스크 기반 테스트(Risk based Testing)

리스크 기반 테스트 기법은 테스트 대상에 비해서 테스트 자원이 부족한 경우 효과적이고, 효율적인 테스트 수행을 위해 적용 될 수 있다. 해당 기법은 크게 리스크 식별과 리스크 분석, 그리고 리스크 계획의 세 단계로 구분 진행된다.

리스크 식별 단계에서는 제품 품질관점에서 테스트 대상이 될 항목을 식별하고, 프로젝트나 제품에 대한 리스크 요소를 식별한다.

리스크 분석 단계에서는 장애 발생가능성과 장애로 인한 영향을 식별하고 리스크 우선순위를 결정한다.

마지막으로 리스크 계획 단계에서는 리스크의 우선순위에 따른 대처 방안 및 완화 정책을 수립하며, 이후 테스트 수행 시 커버리지를 고려하여 선택과 집중을 통해 테스트를 수행하게 된다.

□ 시나리오 테스트

시나리오 테스트 기법은 단일 기능에 대한 결함 여부를 확인하는 것이

아니라, 서로 다른 컴포넌트 사이의 상호작용과 간섭으로 발생할 수 있는 결함을 발견하기 위한 기법이다.

본 테스트에서는 리스크 분석을 통해 ZenLoadBalancer의 기능 및 비기능 항목들에 대한 기능적/ 기술적 아이템을 정의하였다. 또한, 각 정의된 아이템 별 사용자 시나리오를 바탕으로 테스트 아이디어를 도출하였다.

3. 기능 테스트 수행 결과

기능 테스트 수행 관련 세부 케이스 및 상세 내역은 「ZenLoadBalancer 테스트 케이스」 별첨 문서를 참고한다.

□ 탐색적 테스트 현황

기능 아이템	기본 차터	테스트 아이디어
ZenLoadBalancer 기본 환경 구성	1	6
ZenLoadBalancer 관리	3	21
ZenLoadBalancer 모니터링	2	7
ZenLoadBalancer 설정	5	24
ZenLoadBalancer License 정보 관리	1	2
합 계	12	60

[표 5. 테스트 아이디어 현황]

□ 테스트 결과

탐색적 테스트를 통한 테스트 수행 결과 내용은 아래와 같다.

분류		PASS	FAIL	N/T	N/A
기능	테스트 아이디어				
ZenLoadBalancer 기본 환경 구성	6	6	0	0	0
ZenLoadBalancer 관리	21	20	1	0	0
ZenLoadBalancer 모니터링	7	7	0	0	0
ZenLoadBalancer 설정	24	24	0	0	0
ZenLoadBalancer License 정보 관리	2	2	0	0	0

[표 5. 테스트 결과]

VI. 종합

□ ZenLoadBalancer 테스트 수행 결과 공개SW로 구성된 Stack 상에서 기능 테스트 수행 시 Profile 적용 설정 항목관련 이슈 1건 (v3.03기준 / 상세 내역은 테스트 케이스 별첨 문서 참조)을 제외한 제반 기능들이 각 기능 resource들과 유기적으로 동작함을 확인하였다.

□ ZenLoadBalancer는 Linux 기반으로 운영되고, 사용자의 접근이 상대적으로 용이한 웹 UI 상 관리자 환경을 제공한다. 이에 따라 네트워크 환경에 대한 Dashboard 기능 및 클러스터링 구성을 포함 다양한 설정, 관리 기능이 제공된다.

ZenLoadBalancer는 Load Balancing이 필요한 메일 서버를 비롯하여 NAT가 적용되는 기업 별 보안 서버 망 운용에 이르기까지 다양한 클라우드 컴퓨팅 환경에 유용한 솔루션으로 적용될 수 있다.

※ 참고 자료

- [1] <http://www.zenloadbalancer.com/>
- [2] <http://sourceforge.net/p/zenloadbalancer/wiki/Home>
- [3] <http://www.facebook.com/zenlb/timeline>