로드밸런서(VPX) 매뉴얼

목 차

1. 개요 3

1.1 목적 3
1.2 범위 3
1.3 ucloud server 로드밸런서 구동 방식 4

2. 로드밸런서 기술사항 5

2.1 로드밸런서 옵션 (LOAD BALANCING ALGORITHMS) 6
2.2 ucloud server 로드밸런서 특징 7
2.3 TCP OFFLOAD 8

1. 개요

본 문서는 olleh ucloud 의 부가서비스인 로드밸런서 를 사용하는 방법에 대해 문서화 합니다 .

1.1 목적

본 문서는 ucloud server 의 로드밸런서 서비스를 사용할 수 있도록 포탈에서 신청 및 세부 실행 및 셋팅하는 방법을 설명 하는데 목적이 있습니다 .

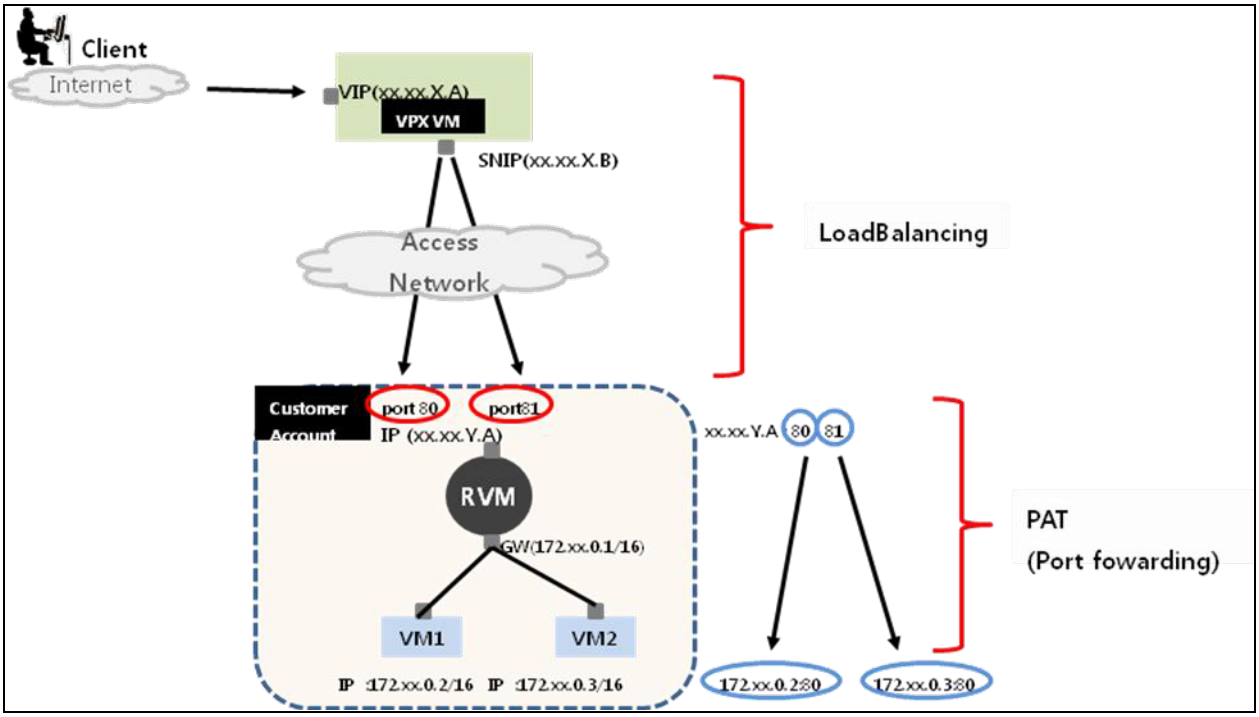
1.2 범위

본 문서에서는 ucloud 환경에서 부가서비스인 로드밸런서 를 사용하여 VM 에 대한 부하분산을 할 수 있도록 설명하고 있습니다 .

이 문서의 사용 범위는 다음과 같습니다 .

- 로드밸런서 신청/해지 및 생성/삭제 방법
- 로드밸런서 를 위한 세팅 방법

1.3 ucloud server 로드밸런서 구동 방식



Load Balancing 은 리소스의 활용을 최적화 하기 위해 client 요청을 여러 server(customer VM)에게 분산시켜 주기 위한 기술입니다. 많은 수의 client 에 서비스를 제공하는 제한된 수의 서버에는 과부하가 발생하고 server 성능이 저하될 수 있습니다. ucloud server 로드밸런서는 요청이 수신되면 각 client 요청을 가장 잘 처리할 수 있는 서버에 전달함으로써 병목 현상을 막을 수 있는 부하 분산 기술을 사용합니다.

그림에서와 같이 Client 가 서비스에 접근을 하기 위해 VPX VM 의 IP 인 VIP 를 통하게 됩니다. VPX VM 은 설정된 Load Balancing option 방식을 통해 Customer Account IP/port(PUBLIC IP/PORT)에 분산됩니다. 각 port 로 분산된 request 는 port forwarding, NAT 기능을 하는 virtual router(RVM)을 통해 각 서비스에 도달하게 되어 Client 의 요청에 응답을 하게 됩니다.

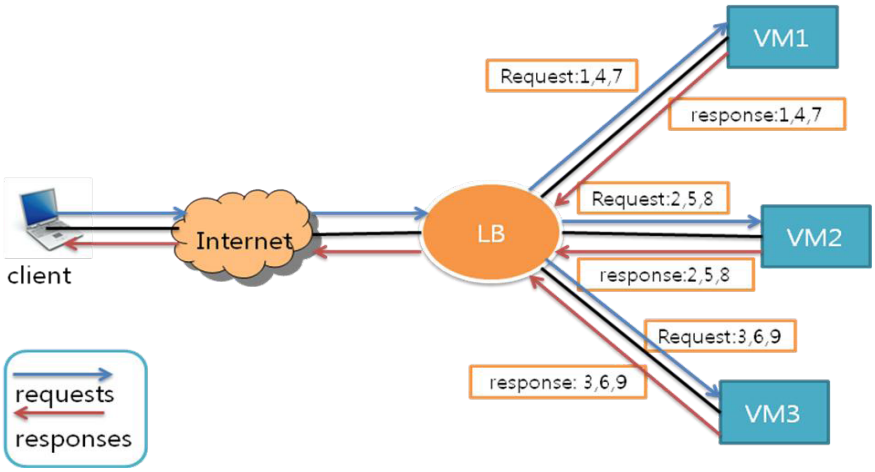
3. 로드밸런서 기술사항

3.1 로드밸런서 옵션 (Load Balancing Algorithms)

- Round robin

  Round robin 방식은 Client의 요청을 순차적으로 순환을 하여 Load Balancing을 하는 방식입니다.

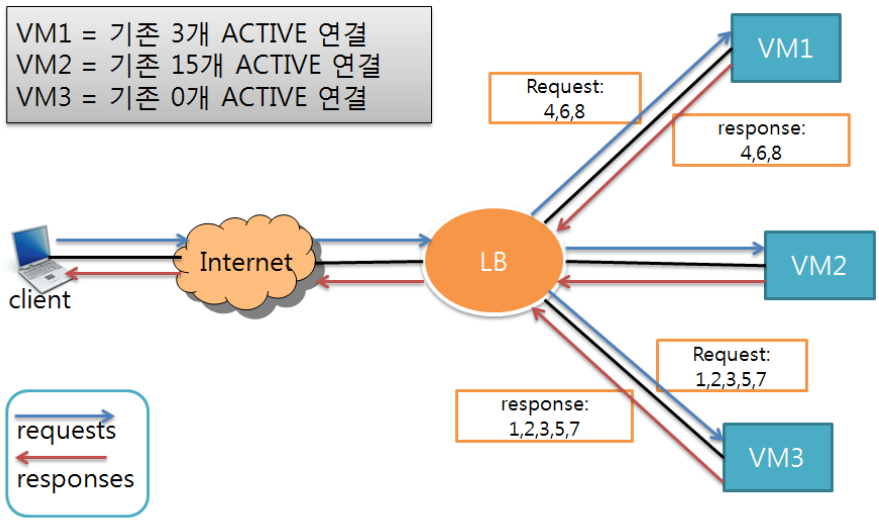
  ▸ Round robin 옵션의 example



- Least Connection

  Least Connection 방식은 Client의 요청을 VPX(LB)와 서버(VM1, VM2, VM3)의 Active 연결이 가장 적은 연결을 선택하여 Load Balancing을 하는 방식입니다.

  ▸ Least Connection 옵션의 example

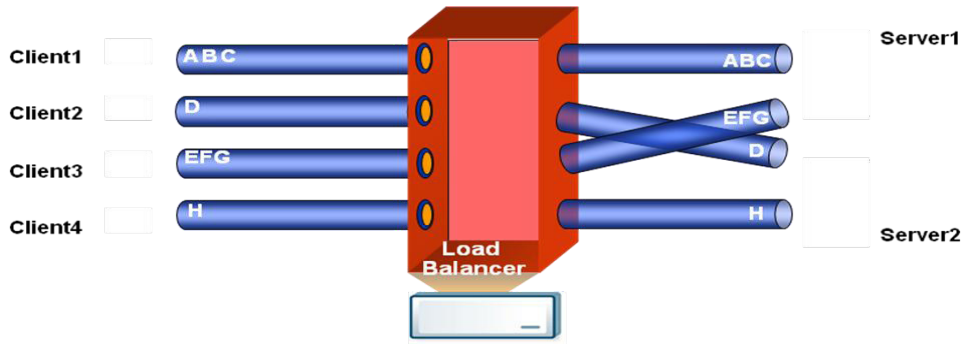


- Hashing(Source IP Hash, Source IP Hash + Port)

  Hash 방식은 client의 Source IP 정보 및 Source IP + Port 정보를 바탕으로 hash한 결과에 의해 load balancing을 합니다. Hash 방식은 동일 정보를 가진 request가 동일 server로 Load Balancing 되도록 할 때 주로 사용합니다.

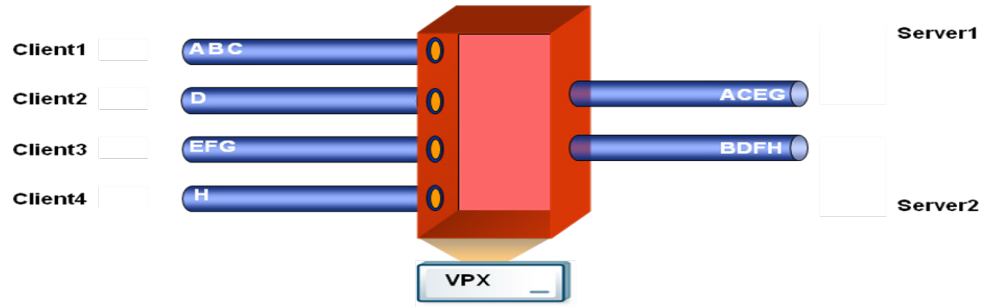
3.2 ucloud 로드밸런서 특징

- 기본 로드밸런서 기술



기본 L4 로드밸런서 기술은 각각의 서버로의 TCP Session 에 대한 로드밸런서 는 정확히 이루어졌으나 실제 사용자가 요청하는 데이터에 대한 로드밸런서 는 이루어 지지 않습니다 .

- ucloud 로드밸런서 기술



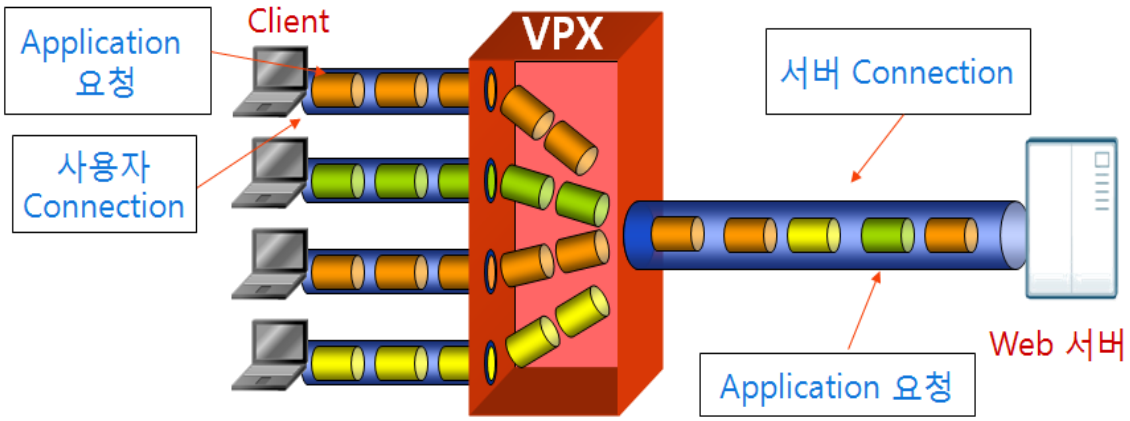
ucloud 로드밸런서 기술은 사용자가 요청한 각각의 데이터들에 대해, 모든 서버로 각각 로드밸런싱 함으로써 , 실제 사용자가 요청한 데이터들은 모든 서버로 정확히 로드밸런서 가 가능해졌습니다 .

- 기본 로드밸런서 와 ucloud 로드밸런서 비교

	기본 로드밸런서	ucloud 로드밸런서
성 능	소형	중대형
동작 방식	기존의 Virtual Router 가 로드밸런서 수행	고객별로 로드밸런서 전용 VM 을 할당
기존 서비스 영향도	높음	낮음
로드밸런서 옵션 선택	가능(3 가지 옵션 선택가능)	가능(4 가지 옵션 선택가능)
네트워크 대역폭 선택	불가	가능(100/300/500Mbps선택가능)
Client 의 Source IP 확인 가능 여부	불가	가능(HTTP 서비스에 한함)
가격	무료	유료(대역폭별 정액 과금)
안정성	낮음	높음(이중화 구성)

3.3 TCP OFFLOAD

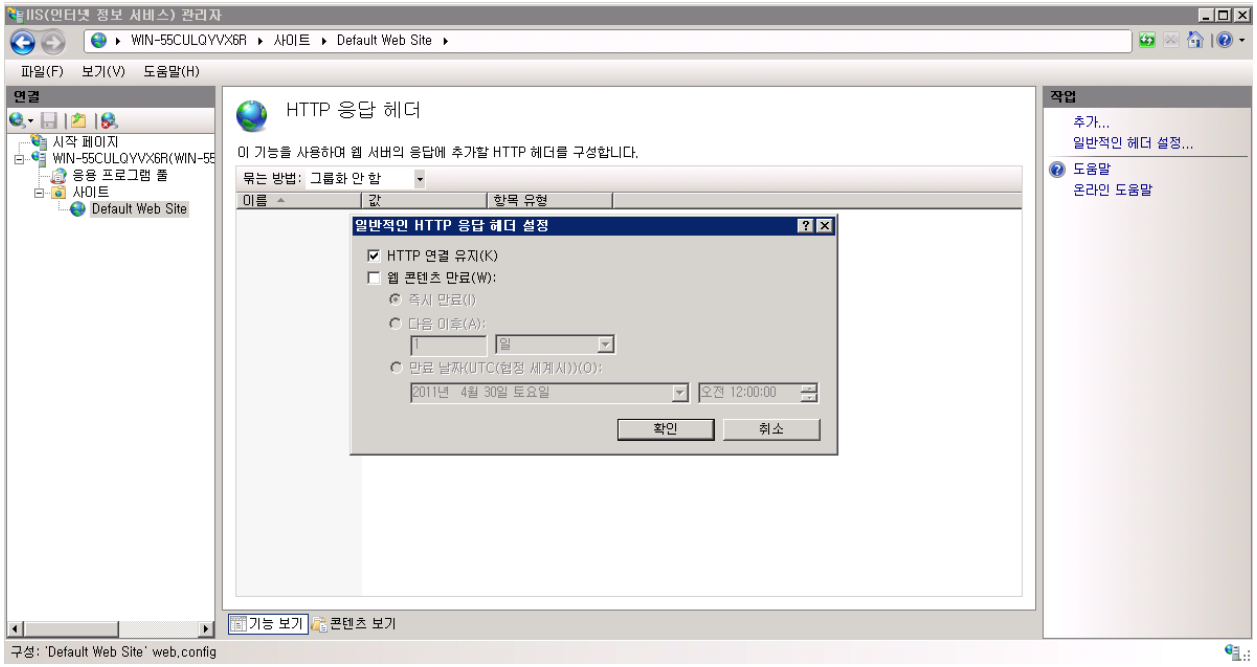
- TCP Multiplexing 을 사용한 TCP Offload



① Client 의 TCP 요청에 VPX 가 server 를 대신하여 응답하여 3-way handshake 진행
② Client 가 Application 에 대한 요청 전송
③ VPX 가 server 와 TCP Connection 생성
④ Client 의 Application 요청을 server 로 전달
⑤ 추가 Client 의 요청 수신 시, VPX 와 server 간의 TCP Connection 이 idle 한 경우 이를 재사용
⑥ Server 와 VPX 간의 Connection 을 통해 다수의 Client 의 요청을 처리

- TCP Multiplexing 사용시 이점
  - Transport Layer 와 application request(Layer7)간의 연관성을 상호 독립시킴으로써 모든 Client 의 요청에 대해 최적화 함
  - Server 또는 Cache 로부터 Transport Layer Processing 을 Offload 시켜 Server 의 CPU 사용률을 획기적으로 경감 시키는 효과가 발생

- TCP Offload를 위한 서버 설정(Aparche2.0웹서버 사용자)
  - /etc/httpd/conf/httpd.conf에서 KeepAlive On 으로 설정
  - 저장 후 데몬 재구동으로 변경사항 적용
  - Aparche2.0웹서버 성능향상을 위한 설정값 변경(권고)

위치	기본값	변경 후
/etc/httpd/conf/httpd.conf	<IfModule prefork.c> StartServers 8 MinSpareServers 5 MaxSpareServers 20 ServerLimit 256 MaxClients 256 MaxRequestsPerChild 4000 </IfModule>	<IfModule prefork.c> StartServers 256 MinSpareServers 50 MaxSpareServers 100 ServerLimit 8192 MaxClients 5000 MaxRequestsPerChild 4000

- TCP Offload를 위한 서버 설정 (IIS 7.0 웹서버 사용자)
  ① 시작 ▶ 관리도구 ▶ IIS 관리자 ▶ Local PC POP Down ▶ 사이트 폴더 ▶ Default Web Site
  ② Http 응답헤더 ▶ (우측상단) 일반적인 헤더설정 ▶ HTTP 연결유지 체크 후 확인