

HA(High Availability, 고가용성)

220307 데일리 콘텐츠

HA(High Availability)

High Availability(고가용성)

- 서버와 네트워크, 프로그램 등의 정보 시스템이 오랜 기간 동안 지속적으로 정상 운영이 가능한 성질
- 시스템에서 이슈 발생 시, 얼마나 빠른 시간 내(유저가 느끼지 못하는 시간)에 복구가 가능한지에 대한 척도
- ex) 카카오톡에서 장애를 겪지 못하는 것은 완벽한 코드를 올려서 서비스를 하는 것이 아니라 HA를 잘 구성한 것임. (서버 분산처리를 잘 하여 한 서버가 다운 됐을 때 해당 서버로 가는 트래픽을 차단 시켜 다른 서버로 우회 시키거나 자동으로 서버를 복구하는 시스템을 구성하는 것 등)



일정 기간내에서 프로젝트를 진행할 때 HA를 고려를 할 경우 시간이 모자랄 수 있음.

HA를 고려하는 것이 **토이프로젝트와 실제 서비스와의 차별점**이 될 수 있음.

HA 관련 용어

Active / Standby

장애가 났을 때, 서버가 떨어졌을 때 복구하기 위한 전략, 안정성.

- Active
 - client로 부터 request를 받아서 처리하는 역할
- Standby
 - 예측한 이벤트(장애 등)가 발생했을 때, Active 대신 request를 처리하는 역할
 - 후의 Backup과 완전 같은 역할

HA(High Availability)



- HA 관련, 용어정리 Active / Standby / Master / Slave / Backup

Active / Standby

Active

client로 부터 request를 받아서 처리하는 역할

Standby

예측한 이벤트(장애 등)가 발생했을 때, Active 대신 request를 처리하는 역할

Standby 서버가 Active 서버가 살아있는지 계속 체크하고 Active가 반응이 없으면 그 자리를 Standby가 대신하여 처리함

ex) 유저 입장에서는, 한 번 접속했을 때 접속이 되지 않았다가 다시 접속하면 되는 경우

Master / Slave / Backup

부하 분산하여 더 많은 트래픽을 처리하기 위한 전략. 여러개의 기능을 수행 하는데 있어 주 기능을 마스터가 수행하고 트래픽이 많이 발생되는 기능을 Slave가 맡아서 수행하도록 함.

- Master
 - 하나의 역할을 수행 하는데 있어 동작의 주체가 되는 역할을 수행
- Slave
 - '주로' 마스터의 지시에 따라 종속적인 역할을 수행
- Backup
 - 특정 서버의 역할을 대체하기 위해 준비된 서버
 - 위의 Standby와 완전 같은 역할

HA(High Availability)



- HA 관련, 용어정리 Active / Standby / Master / Slave / Backup

Master / Slave / Backup

Master

하나의 역할을 수행 하는데 있어 동작의 주체가 되는 역할을 수행

Slave

'주로' 마스터의 지시에 따라, 종속적인 역할을 수행

Backup

특정 서버의 역할을 대체하기 위해, 준비된 서버

트래픽을 많이 발생시키는 것을 Slave가 가져감 ex) CRUD에서의 R

Slave가 양이 많기 때문에 나눌 수록 부하가 적음

Backup은 앞의 Standby와 같은 역할

Active - Standby는 **안정성** 이면 Master - Slave는 **부하 분산**

이중화

Standby가 Active에 Heartbeat를 하여 서버가 살아있는지 계속 확인함. Active가 죽으면 Standby가 그 역할을 대신함.

- Cache server의 다운 발생 시, 대체할 수 있는 미러링 서버를 구성
- Active / Standby 와 Master / Backup 구성



미러링: 데이터를 복제하는 것

fail over: Active가 죽어서 Standby가 Active로 올라오는 것을 뜻함.

auto fail over: fail over를 자동으로도 이루어지도록 한 것. auto로 진행되는 것을 지향함. Redis에서 Sentinel이라는 솔루션을 제공하여 fail over(standby-active)를 자동화 시켜줌.

Redis Sentinel의 auto fail over: Redis Sentinel에서는 auto fail over과정에서 투표를 하기 때문에 Redis를 홀수개를 올려야함(최소 3개).

RDB

- Master와 Slave는 dedicated storage를 사용하여 async 방식으로 데이터 복제(실제로 데이터를 복제할 지 말지는 선택적)
- Master에서 CRUD의 CUD를 담당하고, Slave에서 R을 담당
- Slave 서버는 Scale-out이 쉽게 가능 (1:N)

HA(High Availability)

- HA 관련, 용어정리 Active / Standby / Master / Slave / Backup

예) MySQL에서 Replication 구성

- Master와 Slave는 dedicated storage 사용하여 async 방식으로 데이터 복제
- Master에서 CRUD의 CUD를 담당하고, Slave에서 R을 담당
- Slave 서버는 Scale-out이 쉽게 가능 (1:N)

- shard: DB에서 분산을 한 것
- Scale out: 서버를 2대를 3대로 늘린 것, 상대적으로 경제적, 단 관리 포인트가 늘어남. 미래에 있을 Scale out을 고려하고 설계를 하는 것이 좋음
- Scale up: 스펙을 업 시키는 것(메모리를 늘리거나, CPU 코어 수를 늘리는 것 등), 한계가 있음

Load Balancer

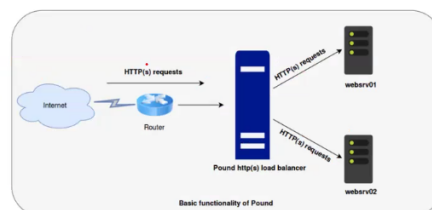
- 부하 분산의 목적
- n개의 Active Server로 구성
- 모든 Server가 Active

HA(High Availability)

- HA 관련, 용어정리 Active / Standby / Master / Slave / Backup

예) Load Balancer 아래에 n개의 server 구성

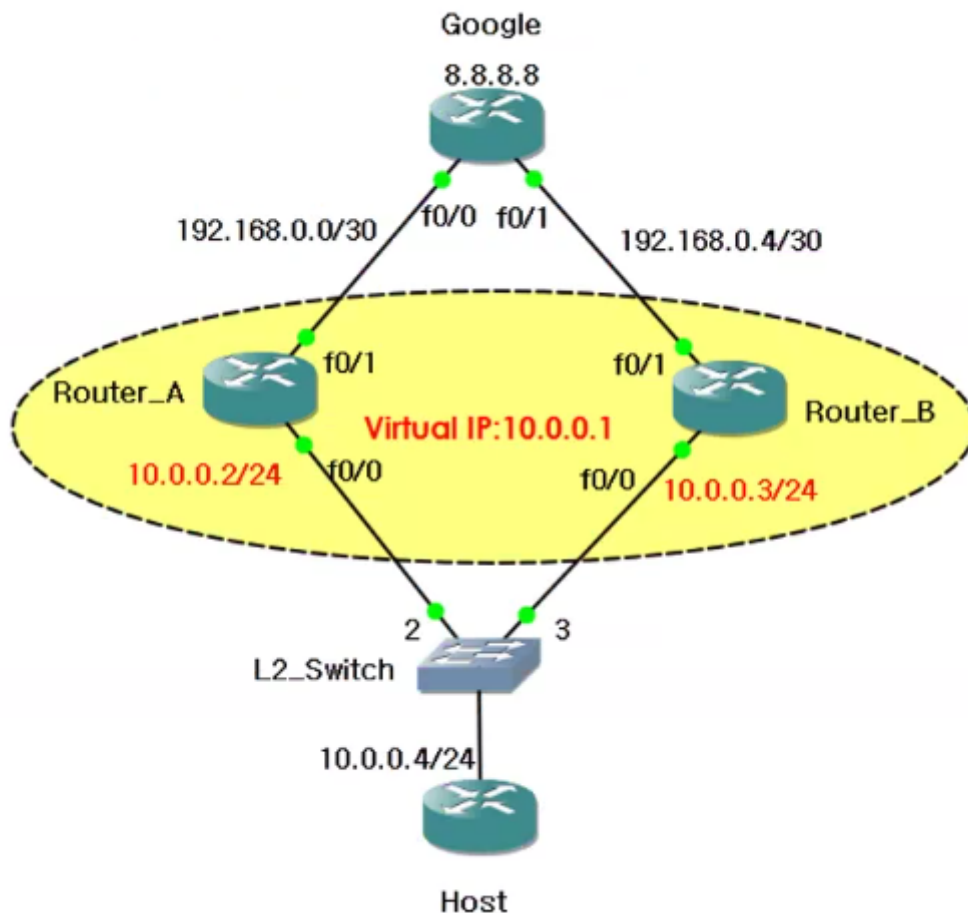
- 부하분산의 목적
- n개의 Active Server로 구성
- 모든 Server가 Active



- load balancing 하는 방법은 여러 알고리즘이 있음, load balancer를 직접 구축하는 경우는 거의 없고 물리장치를 사용함

VRRP(Virtual Router Redundancy Protocol)

- Master Router와 Backup Router로 구성
- Master Router가 '주'로서 동작하고, Master의 장애 시 Backup Router가 대체
- Active / Standby 와 Master / Backup 구성



멀캠에서 노트북(Host)을 켜서 구글에 접속하려고 하면 라우터(Router_A)를 타고 나감. 구글 입장에서는 Virtual IP로 요청을 받고, 응답을 라우터에 보냄. 응답을 받은 라우터는 헤더에 사설 IP 값이 있어서 이걸 다시 사설 망으로 보내줌. 위와 같은 시스템을 통해 사용자는 사설IP를 통해서 인터넷을 한다고 보면 됨.

라우터 A가 master, 라우터 B가 backup이며 master가 죽었을 경우 라우터 B가 이 역할을 대신함.

- 라우터: 공인 IP를 가진 unique한 장비.
- 퍼블릭망: 라우터가 존재하는 노란 구역
- 사설망: 노란 구역 하단 부분
- 퍼블릭망과 사설망 사이에는 ACL(access control list)라는 방화벽이 존재함
- 아웃 바운드(out bound): 사설망에서 퍼블릭망으로 가는 것을 말함
- 인 바운드(in bound): 퍼블릭망에서 사설망으로 가는 것을 인 바운드라고 함

정리: HA를 고려/반영했다(트래픽에 대해 고려했다)는 것은 굉장히 플러스 요소가 됨, 무조건 알아야 되는 상식인데 가르쳐주는게 굉장히 어려운 내용이기 때문

