1.네트워크_로드밸런싱

1. 정의
    - 한 서버에 트래픽이 집중되는 것을 방지하고, 여러 서버에 적절히 분산시키는 기술
2. 서버를 확장하는 방법
    a. Scale up
    b. Scale out
3. Scale up(수직 확장)
    - 서버 자체 사양을 높이는 것
    - 방법
        - 디스크, CPU, RAM 증설 또는 클라우스 인스턴스 사양 업그레이드
        - 기존 서버의 자원이 한계에 도달한 경우 하드웨어적으로는 서버에 디스크를 추가하거나 CPU나 RAM과 같은 자원을 업그레이드
        - 소프트웨워적으로는 AWS와 같은 클라우드 서비스의 인스턴스 사양을 높임
    - 장점
        - 추가 네트워크 연결 없이 자원 확장 가능
        - 관리 비용/운영 이슈가 적고 확장이 간단
        - 서버가 1대 이ㅡ로 데이터 일관성 유지
    - 단점
        - 하드웨어 업그레이드 비용 부담이 큼
        - 확장에 한계 존재
        - 단일 서버 장애 시 서비스 전체 장애로 이어짐
4. Scale out(수평 확장)
    - 방법
        - 서버의 개수를 늘려 트래픽을 분산하는 방식
        - 클라우드 환경에서는 Auto Scaling으로 자동 증설 가능

    - 기존 서버의 자원이 한계에 도달한 경우 비슷한 사양의 서버를 추가로 연결해 트래픽을 분산시켜 기존 서버의 부하를 줄임
    - Horizontal scaling이라고도 불림
  - 장점
    - 유연한 확장성 확보
    - 일부 서버 다운 시에도 서비스 유지 가능
    - 필요한 만큼 서버 증설 가능
  - 단점
    - 병렬 분산 아키텍처에 대한 높은 이해도 요구
    - 데이터 일관성 문제 발생 가능
    - 로드 밸런서 필요
5. 로드 밸런서
  - 정의
    - 클라이언트와 서버 사이에서 서버의 부하를 분산시키는 장치(하드웨어/소프트웨어)
  - 기능
    - 네트워크 트래픽 및 요청을 적절히 분배
    - 정상 서버로만 요청 전달
    - 서비스 중단 없이 서버 추가/제거 가능
  - 종류
    - L4 Load Balancer(전송 계층 기반)
    - L7 Load Balancer(응용 계층 기반)
6. L4 Load Balancer
  - 전송 계층(4계층, TCP/UDP) 기반
  - IP주소, 포트 번호를 활용해 부하 분산
  - 특징
    - 빠르고 효율적
    - 데이터 내용을 열어보지 않아 안전

    - 비용이 저렴
  - 단점
    - 사용자의 IP가 자주 변경되면 연속 서비스 제공이 어려움
  - 한 대의 서버가 각각 다른 포트 번호를 부여하여 다수의 서버 프로그램을 운영하고 있다면 최소 L4 이상의 로드 밸런서를 사용 해야 함
7. L7 Load Balancer
  - 응용 계층(7계층, HTTP 등) 기반
  - 요청 정보(HTTP 헤더, 쿠키, URL 등)에 따라 세밀한 분산 가능
  - 특징
    - URL, 헤더, 쿠키 값에 따라 트래픽 분산
    - 서버 응답까지 확인 및 분석 가능
    - 특정 패턴의 공격(예: DDos) 방어 가능
  - 보안 및 트래픽 제어 측면에서 확용도 높음
  - 서버의 응답까지 확인하고 분석할 수 있다는 특징을 가짐
8. 로드 밸런싱 알고리즘
  - 라운드 로빈
  - 가중 라운드 로빈
  - IP 해시
  - 최소 연결
  - 최소 응답 시간
9. 라운드 로빈
  - 로드 밸런싱에서 가장 많이 사용되는 알고리즘
  - 클라이언트의 요청을 각 서버에 순차적으로 분배
  - 들어온 요청을 빠르게 서버로 분산하는 작업에 포커스를 맞춤
  - 모든 서버의 스펙이 동일한 경우 적합
10. 가중 라운드 로빈
  - 서버별 가중치 부여 → 높은 가중치 서버에 더 많은 요청 전달
  - 특정 서버의 스펙이 더 좋을 경우 가중 라운드 로빈 알고리즘 사용

- 예시
    - A 서버의 가중치가 3이고 B 서버의 가중치가 1일 경우, 8개의 요청을 받았다면 A 서버로는 6개 B 서버로는 2개의 요청이 설정된 가중치를 기준으로 전달

11. IP 해시
    - 클라언트 IP 기반으로 특정 서버에 요청을 할당
    - 동일 사용자는 항상 동일 서버로 연결
    - 세션 클러스터링이 없을 때 주로 사용
    - 특정 주소에서 접속량이 많을 경우 편함
        - 세션 클러스터링 : 2대 이상의 WAS 또는 서버를 사용할 때 로드 밸런싱, 장애 대비 등 세션을 공유
12. 최소 연결
    - 현재 연결 수가 가장 적은 서버에 요청 전송
    - 트래픽이 균등하지 않은 경우 적합
13. 최소 응답 시간
    - 연결 수와 응답 시간을 모두 고려해 가장 빠른 서버에 요청 전송
14. 소프트웨어 기반 로드 밸런싱
    - 기본적으로 Reverse Proxy를 기반으로 동작
    - 클라이언트 요청을 내부 서버에 전달하고 결과를 반한
    - 대표 오픈 소스
        - Nginx: 경량, 높은 성능의 리버스 프록시/로드 밸런서
        - HAProxy: 다양한 로드밸런싱 기능과 고가용성 지원