1. 파트1: 확장성 활성화
   1. 확장성 활성화 모범사례
      1. 수요 변화에 **신속하게 대응**할 수 있는 능력
      2. 예) 경보 임계값에 도달한 서버는 Auto Scaling이 알림을 받고 새서버를 준비
   2. 조정의 비교
      1. 수직적 조정
         * 스케일 업 혹은 다운
         * 예) 인스턴스의 사양 변경
         * 대표적으로 DB가 이에 해당
      2. 수평적 조정
         * 스케일 인 혹은 다운
         * 예) 인스턴스의 수 변경
         * 대표적으로 웹 서버가 이에 해당
2. 파트2: 조정이 필요한지 어떻게 알 수 있습니까?
   1. 지표 수집: Amazon CloudWatch
      1. 인스턴스를 모니터링
      2. 알림을 전송, Auto Scaling 작업 트리거
      3. 다음과 같은 사항을 제공
         * 분산 통계 수집 시스템
         * 하이퍼바이저 수준에서 원활하게 수집되는 지표
         * 사용자 정의 지표를 생성 및 수집하는 기능 (요금 부과)
   2. CloudWatch Log로 로그 파일 저장 및 모니터링
      1. 관리자 및 기타 관계자는 CloudWatch 로그 열람 가능
      2. S3에 저장되어 다른 서비스나 사용자가 접근 가능
3. 파트3: Auto Scaling
   1. Auto Scaling
      1. 인스턴스를 시작 혹은 종료
      2. **새 인스턴스**를 로드 밸런서에 자동 등록
      3. **여러 가용 영역에 걸쳐** 시작 가능
   2. 수요의 변화에 대응하고 유연한 시스템을 구축!
   3. 작동 방법
      1. 시작 구성: 인스턴스를 시작하는 방법이나 속성을 정의
      2. Auto Scaling 그룹: 인스턴스를 시작할 때, 자동으로 정보 부여
      3. Auto Scaling 정책: 경보를 기준으로 작동 (예측 가능한 경우)
         * Cooldown 매개변수: 종료 혹은 시작 전에 시간을 가짐.
      4. 예약된 작업: 특정 시점에 조정 작업을 수행 하라고 지시 (예측불가)
   4. 최소 용량 크기 결정
      1. 원하는 용량: 유지 용량, 장애가 생겼을 때
      2. 최소 용량: Scale in에서의 최소 용량
         * AZ 별로 최소한 1개의 인스턴스로 시작해야 하는 경우. 최소 용량 크기는 AZ수로 설정
      3. 최대 용량: Scale out에서의 최대 용량
   5. 고려 사항
      1. Scale in 할 때 더 주의를 기울이고, 공격적인 인스턴스 종료는 피함
      2. 미리 확장하고, 천천히 축소함
      3. 최소 및 최대 용량 매개변수 값은 신중하게 설정
      4. 수명주기 후크를 사용
         * 인스턴스를 시작 또는 종료됨에 따라 사용자 정의 작업을 수행
      5. 상태 저장 App의 경우, Auto Scaling 그룹에서 시작되는 인스턴스에 대한 추가 자동 구성 필요
4. EC2 자동 복구
   1. EC2 자동 복구 기능을 사용해 손상된 인스턴스를 자동으로 교체함
   2. CloudWatch에서 탐지
   3. 교체 인스턴스
      1. 동일한 인스턴스 ID/ 메타데이터, IP 주소 유지
      2. 손상된 인스턴스의 인 메모리 데이터는 사용할 수 없음
   4. EBS 지원 스토리지만 사용해야 함
5. 데이터 스토어 조정
   1. **RDS의 성능을 높이는 방법(꼭 기억!!)**
      1. 인스턴스 유형을 높임
      2. 디스크 성능을 높임 (프로비저닝 🡪 RAID)
      3. 읽기 복제본 만듦 (복제는 비동기식)
      4. RDS 앞에 ElastiCache를 배치
   2. 샤딩으로 RDS 쓰기 조정
      1. 샤딩: 데이터를 **큰 쳥크**로 분할
      2. **뛰어난 성능**과 **높은 운영 효율성**을 제공
   3. DynamoDB용 Auto Scaling
      1. 용량 관리 자동화
      2. 쓰기 용량에 대한 상한 및 하한 제공
      3. CloudWatch 경보를 제공하여 처리량 소비 모니터링
      4. 모든 새 테이블 및 색인에 대한 기본값
6. Lambda 및 이벤트 중심의 조정
   1. Lambda
      1. 이벤트 혹은 시간 기반 간격에 대한 응답으로 **상태 비저장 코드**를 실행하는 **완전 관리형 컴퓨팅 서비스**
      2. 할 수 있는 작업
         * 자체 코드 사용 가능
         * 코드 병렬 실행
         * 백엔드, 이벤트 핸들러, 데이터 처리 시스템 생성
         * 유휴 리소스에 대해 비용을 지불할 필요가 없음
   2. 어떻게 조정에 사용되는가?
      1. 조정 이벤트가 Lambda 함수를 트리거 할 수 있음
      2. Lambda에서 실행되는 코드는 AWS API에 액세스 할 수 있고, IAM역할을 통해 권한을 부여될 수 있음
      3. **다른 AWS서비스에 대한 API 호출을 자동으로 수행**
      4. 컨테이너 기반 인스턴스 조정(Docker, Amazon Elastic Container Service 등)

<Docker 예>

