1. 파트 2: Amazon Elastic Compute Cloud (Amazon EC2) 소개
   1. Amazon EC2: Elastic Compute Cloud
      1. 하이퍼 바이저 위에 서버 이미지를 올려 사용
      2. 인스턴스 (가상 컴퓨팅 환경)를 기반으로 운영
         * 가상머신 = 가상서버 = 인스턴스
         * 자체 서버 이미지 사용
         * 필요할 때 추가 / 삭제 가능
         * 코어 수 / 메모리 크기 / 스토리지 크기 및 유형 등 정의됨
         * OS와 그 설정 & 패치의 초기 상태 & 응용프로그램 혹은 시스템 응용프로그램: x86 / Linux 기반
      3. Amazon Machine Image (AMI): 인스턴스에 초기 설치될 SW 정보
      4. Amazon EC2 요금
         * 온디멘드 - 최대 가격
           1. 필요할 때 인스턴스 사용
           2. TEST나 사용량이 측정되지 않을 때
         * 스팟인스턴스 - 중간 가격
           1. 기준가격을 두고, 최대가격보다 높으면 종료
           2. 클라우드의 여유공간을 활용하기에 비교적 저렴
           3. 대규모로 중간에 멈춰도 상관없는 인스턴스를 운영할 때 => 대규모로 상태 비저장을 요구하는 워크로드
         * 예약인스턴스 – 낮은 가격
           1. 일정 수준을 예약. 확인
           2. Web Server등 지속적으로 연결되어야 하는 인스턴스
         * 전용 호스트 – 낮은 가격
           1. 물리적으로 다른 서버와 격리
           2. ex) 미 국방성
2. 파트 4: Amazon EC2 비용 최적화
   1. 비용 최적화: 필요하거나 필요한 것만 비용을 지불
      1. 기존의 온디멘드 환경을 최적화를 수행(Lift and Shift)하여 비용을 줄인다.
         * ex) EC2에서 쓰지 않는 인스턴스는 삭제
      2. 최적 사이징: 인스턴스 Type 결정
      3. 예약 인스턴스: 항상 가동하는 자원이 70~80%정도 유지하도록 설정!
      4. 탄력성 향상: 필요할 때 인스턴스를 사용하고, 필요 없으면 종료
      5. 모니터와 향상
         * Cost Explore: 비용에 대한 정보 제공 서비스
         * 인스턴스의 태그를 기반으로 정렬하여 비용 분석 및 최적화 가능
3. 파트 5: AWS Lambda 소개
   1. AWS Lambda의 특징
      1. 완전 관리형 서버리스 컴퓨팅
      2. 이벤트 중심 실행
      3. 초 미만 단위 미터링
      4. 최대 5분까지 함수 실행
      5. PaaS (Platform as a Service)
   2. 관리형 vs 비관리형
      1. 관리형: SW적인 부분까지 Amazon에서 관리 / 사용자의 권한 X (PaaS이상)
      2. 비관리형: 물리적 자원만 관리 / 사용자의 권한 O (IaaS만)
   3. Lambda 실행 과정
      1. AWS Lambda로 코드 업로드
      2. 코드를 설정해, 다른 곳에서 트리거 함
      3. Lambda는 트리거 될 때 코드를 실행
      4. 사용한 컴퓨팅 시간에만 요금 청구
4. 파트 6: AWS Elastic Beanstalk 소개 (PaaS)
   1. AWS Elastic Beanstalk
      1. 웹 응용 프로그램을 구동하는 용도
      2. 단독 실행 시, 과금 부과 X
      3. 관리의 복잡성 감소
      4. 응용 프로그램의 배포 과정을 단순화
   2. AWS Elastic Beanstalk와 AWS Lambda의 차이점
      1. AWS Elastic Beanstalk: 웹 응용 프로그램을 올리면 실행해주는 컴퓨팅
      2. AWS Lambda: 이벤트를 수행하는 완전 관리형 서버리스 컴퓨팅