

# AWS 데이터 분석 / 머신러닝

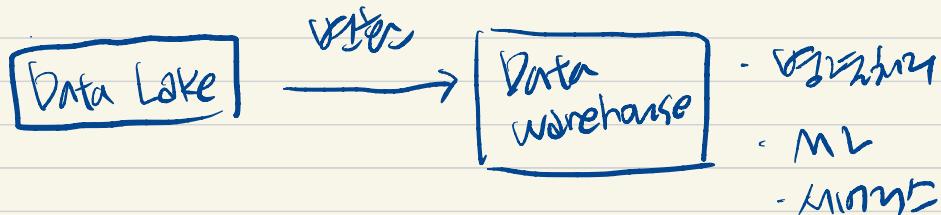
1. 한 회사에 Amazon S3 데이터 레이크가 있습니다. 이 회사에는 데이터 레이크의 데이터를 변환하여 매일 데이터 웨어하우스에 로드하는 솔루션이 필요합니다. 데이터 웨어하우스에는 대량 병렬 처리(MPP) 기능이 있어야 합니다.

그런 다음 데이터 분석가는 데이터에 SQL 명령을 사용하여 머신 러닝(ML) 모델을 만들고 학습해야 합니다. 이 솔루션은 가능한 한 서비스 AWS 서비스를 사용해야 합니다.

어떤 솔루션이 이러한 요구 사항을 충족할까요?

- (A) 매일 Amazon EMR 작업을 실행하여 데이터를 변환하고 Amazon Redshift에 데이터를 로드합니다. Amazon Redshift ML을 사용하여 ML 모델을 만들고 학습합니다.
- B. 매일 Amazon EMR 작업을 실행하여 데이터를 변환하고 Amazon Aurora Serverless에 데이터를 로드합니다. Amazon Aurora ML을 사용하여 ML 모델을 만들고 학습합니다.
- C. 매일 AWS Glue 작업을 실행하여 데이터를 변환하고 Amazon Redshift Serverless에 데이터를 로드합니다. Amazon Redshift ML을 사용하여 ML 모델을 만들고 학습합니다.
- D. 매일 AWS Glue 작업을 실행하여 데이터를 변환하고 Amazon Athena 테이블에 데이터를 로드합니다. Amazon Athena ML을 사용하여 ML 모델을 만들고 학습합니다.

ML 가능 없음



## Data lake

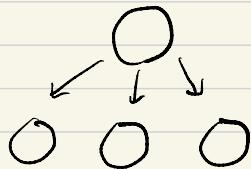
모든 규모의 **структу리화** / **비구조화** 데이터를, 저장할 수 있는 **중앙 집중식** 기관화된 데이터베이스

### c.f. 데이터 워크숍

• 동종부서별 시장영과 업무내적 활용방법으로 충돌이나 간섭을 일으킬 **개별화된 데이터베이스**를 통합화하는 형태

→ **구조화된 데이터에** 대한 접근

## MPP (대량 병렬 처리)



## Amazon EMR (Hadoop / Spark / Flink)

Hadoop, Apache Spark 및 같은批处理 데이터프로세스를  
작동하는 대량 데이터를 처리하고 분석하는 데 사용되는 분산 컴퓨팅

프레임워크

서비스 X 클러스터 관리 필요

## AWS RedShift

구조화 데이터베이스

Glue : ETL API

MVC

Angular.js MVC

Angular.js / Angular.js API Service API . Web API

## 데이터 스트리밍 아키텍처

2. 한 회사가 여러 위치에 데이터 수집 센서를 두고 있습니다. 데이터 수집 센서는 회사에 대량의 데이터를 스트리밍합니다. 이 회사는 AWS에서 대량 스트리밍 데이터를 수집하고 처리하는 플랫폼을 설계하고자 합니다. 이 솔루션은 확장 가능해야 하며 거의 실시간으로 데이터 수집을 지원해야 합니다. 이 회사는 향후 보고를 위해 Amazon S3에 데이터를 저장해야 합니다.

어떤 솔루션이 이러한 요구 사항을 가장 적은 운영 오버헤드로 충족할까요?

- A Amazon Kinesis Data Firehose를 사용하여 스트리밍 데이터를 Amazon S3로 전송합니다.
- B. AWS Glue를 사용하여 스트리밍 데이터를 Amazon S3에 전송합니다.
- C. AWS Lambda를 사용하여 스트리밍 데이터를 전달하고 해당 데이터를 Amazon S3에 저장합니다.
- D. AWS Database Migration Service(AWS DMS)를 사용하여 스트리밍 데이터를 Amazon S3로 전송합니다.

AWS Kinesis

- data Streams : 실시간 분석
- data Firehose : 데이터 수집
- data analytics : 스트리밍 데이터 분석

Glue: 템스 마이트

Lambda: 대량 처리 가능성이 높아짐

DMS: DB와 같은 서비스 마이그레이션

DMS

: 2110121 211012 21101210112

3. 한 회사가 백엔드 프로세서에 점수 업데이트를 스트리밍한 다음 리더보드에 결과를 게시하는 모바일 게임을 개발하고 있습니다. 솔루션 아키텍트는 대규모 트래픽 급증을 처리하고, 자신 순서대로 모바일 게임 업데이트를 처리하고, 처리된 업데이트를 고가용성 데이터베이스에 저장할 수 있는 솔루션을 설계해야 합니다. 또한 이 회사는 솔루션을 유지 관리하는 데 필요한 관리 오버헤드를 최소화하려고 합니다.

솔루션 아키텍트는 이러한 요구 사항을 충족하기 위해 무엇을 해야 할까요?

### 실시간 업데이트 / 순서보장 (삭제 대상)

- A. Amazon Kinesis Data Streams에 점수 업데이트를 푸시합니다. AWS Lambda로 Kinesis Data Streams에서 업데이트를 처리합니다. 처리된 업데이트를 Amazon DynamoDB에 저장합니다. ~~고가용성 DB~~
- B. Amazon Kinesis Data Streams에 점수 업데이트를 푸시합니다. Auto Scaling을 위해 설정된 Amazon EC2 인스턴스 플릿으로 업데이트를 처리합니다. 처리된 업데이트를 Amazon Redshift에 저장합니다. ~~적용X~~
- C. Amazon Simple Notibcation Service(Amazon SNS) 토픽에 점수 업데이트를 푸시합니다. AWS Lambda 함수를 SNS 토픽에 구독하여 업데이트를 처리합니다. 처리된 업데이트를 Amazon EC2에서 실행되는 SQL 데이터베이스에 저장합니다. ~~SNS 순서보장 X~~
- D. Amazon Simple Queue Service(Amazon SQS) 대기열에 점수 업데이트를 푸시합니다. Auto Scaling이 있는 Amazon EC2 인스턴스 플릿을 사용하여 SQS 대기열의 업데이트를 처리합니다. 처리된 업데이트를 Amazon RDS Multi-AZ DB 인스턴스에 저장합니다.

~~SQS 순서보장 X~~

분석용  
전회용 X

- ① 4. 한 회사에서 UDP를 사용하는 수천 개의 지리적으로 분산된 원격 장치에서 데이터를 수신하는 애플리케이션을 실행합니다. 이 애플리케이션은 데이터를 즉시 처리하고 필요한 경우 장치로 메시지를 다시 보냅니다. 데이터는 저장되지 않습니다.  
이 회사는 장치에서 데이터 전송에 대한 지연 시간을 최소화하는 솔루션이 필요합니다. 이 솔루션은 또한 다른 AWS 지역으로의 빠른 장애 조치를 제공해야 합니다.

어떤 솔루션이 이러한 요구 사항을 충족할까요?

### (DNS 서비스)

- A. Amazon Route 53 장애 조치 라우팅 정책을 구성합니다. 두 리전 각각에 네트워크 로드 밸런서(NLB)를 만듭니다. NLB를 구성하여 AWS Lambda 함수를 호출하여 데이터를 처리합니다.
- B. AWS Global Accelerator를 사용합니다. 두 리전 각각에 엔드포인트로 Network Load Balancer(NLB)를 만듭니다. Fargate 시작 유형으로 Amazon Elastic Container Service(Amazon ECS) 클러스터를 만듭니다. 클러스터에서 ECS 서비스를 만듭니다. NLProcess의 대상으로 ECS 서비스를 설정합니다. Amazon ECS에서 데이터를 처리합니다.
- C. AWS Global Accelerator를 사용합니다. 두 리전 각각에 엔드포인트로 Application Load Balancer(ALB)를 만듭니다. Fargate 시작 유형으로 Amazon Elastic Container Service(Amazon ECS) 클러스터를 만듭니다. 클러스터에서 ECS 서비스를 만듭니다. ECS 서비스를 ALB의 대상으로 설정합니다. Amazon ECS에서 데이터를 처리합니다.
- D. Amazon Route 53 장애 조치 라우팅 정책을 구성합니다. 두 리전 각각에 Application Load Balancer(ALB)를 만듭니다. Fargate 시작 유형으로 Amazon Elastic Container Service(Amazon ECS) 클러스터를 만듭니다. 클러스터에서 ECS 서비스를 만듭니다. ECS 서비스를 ALB의 대상으로 설정합니다. Amazon ECS에서 데이터를 처리합니다.

개성으로  
구성

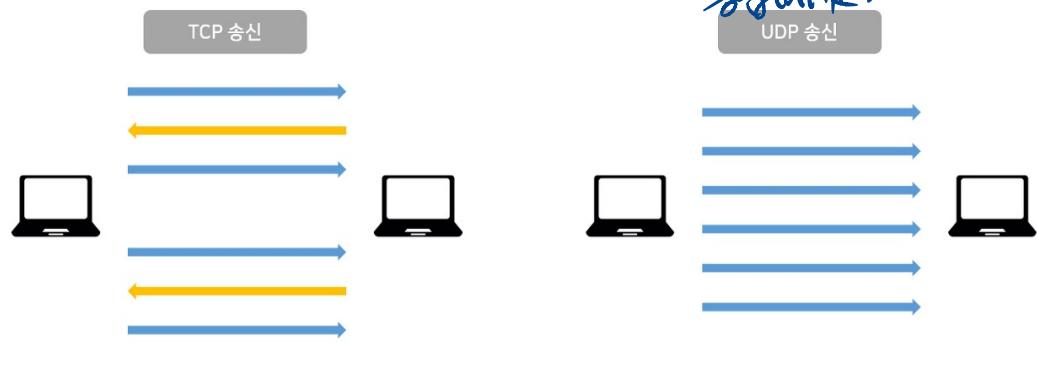
NLB: UDP지원

ALB: HTTPS(S)  
만 가능

Global Accelerator : 전세계로 확장  
구성 가능한 애플리케이션 트래픽 처리  
설정 및 관리

네트워크 기반으로 워커와 사용 가능한 장소 엔드포인트로  
구성 가능한 애플리케이션 엔드포인트로 장애 처리

# TCP / UDP 프로토콜



· TCP: 고정적 단위로 통신하는 방식  
기반(동기화)

→ 데이터 전송에 차이점이 있는 경우 예상

UDP: 비동기적 단위로 통신하는 방식.  
기반(동기화 X)  
기록할 예상과

## 503 : 애플리케이션 과부화 → 트래픽 풀링

5. 한 회사에는 타사 공급업체에서 거의 실시간으로 데이터를 수신할 수 있는 REST 기반 인터페이스가 있는 애플리케이션이 있습니다. 수신되면 애플리케이션은 데이터를 처리하고 추가 분석을 위해 저장합니다. 애플리케이션은 Amazon EC2 인스턴스에서 실행 중입니다. 타사 공급업체는 애플리케이션에 데이터를 보낼 때 많은 503 서비스 사용 불가 오류를 수신했습니다. 데이터 볼륨이 급증하면 컴퓨팅 용량이 최대 한도에 도달하고 애플리케이션은 모든 요청을 처리할 수 없습니다.

솔루션 아키텍트는 확장성이 더 뛰어난 솔루션을 제공하기 위해 어떤 설계를 권장해야 합니까?

A. Amazon Kinesis Data Streams를 사용하여 데이터를 수집합니다. AWS Lambda 함수를 사용하여 데이터를 처리합니다.

B. 기존 애플리케이션 위에 Amazon API Gateway를 사용합니다. 타사 공급업체에 대한 할당량 제한이 있는 사용 계획을 만듭니다. → 배포 및 보호이나 재설정과 유관

C. Amazon Simple Notification Service(Amazon SNS)를 사용하여 데이터를 수집합니다. EC2 인스턴스를 Application Load Balancer 뒤의 Auto Scaling 그룹에 넣습니다.

D. 애플리케이션을 컨테이너로 다시 패키징합니다. Auto Scaling 그룹이 있는 EC2 시작 유형을 사용하여 Amazon Elastic Container Service(Amazon ECS)를 사용하여 애플리케이션을 배포합니다.

Auto Scaling: 서버 사이즈 자동으로 조정하는 서비스

컴퓨팅 파워를 늘리는 방법

05개월 입 0 → 0 ↗ ↙ 스케일 이유 0 → 000

ECS: 컨테이너의 배포, 관리, 확장, 네트워킹 자동화