

19기 Engineering Base

데이터 파이프라인 실습.

18기 엔지 김나연



CONTENTS

01 데이터 파이프라인?

02 구성 요소

- Hadoop
- Spark
- Kafka
- ELK/EFK
- 클라우드 서비스 공급자별 서비스

03 파이프라인 예시

04 실제 데이터 파이프라인 분석해보기

- 7000만 학생이 가입한 인공지능 AI 수학 공부앱, [칸다]
- 여행의 모든 것, 한번에 쉽게 [야놀자]
- 전 세계 1억 8천만 명 이상의 LINE 사용자를 대상으로 하는
글로벌 광고 플랫폼, [LINE Ads]

06 데이터 출처

데이터 파이프라인?

ETL?

지금까지 배운 것들을
종합해보자...

네?

구성 요소

Hadoop

Spark

Kafka

클라우드 서비스 공급자별 서비스



- > 적당한 성능의 범용 컴퓨터 여러 대를 클러스터화하고 큰 크기의 데이터를 클러스터에서 병렬로 동시에 처리하여 처리 속도를 높이는 것을 목적으로 하는 분산처리를 위한 오픈소스 프레임워크

핵심 종류

- **HDFS(분산 데이터 저장):** 하둡 네트워크에 연결된 기기의 데이터를 저장하는 분산형 파일 시스템. 여러 기계(서버)에 대용량 파일을 중복해서 저장함으로써 데이터 안정성을 얻는다.
- **MR(분산 처리):** 대규모 집합으로 매핑한 다음 필터링하여 특정 결과를 찾아내는 방식으로 데이터 처리
- **Yarn:** 리소스 관리 및 일정 예약
- **Zookeeper:** 분산환경에서 서버 간의 상호 조정이 필요한 다양한 서비스를 제공하는 시스템. 분산 동기화를 제공하고 그룹 서비스를 제공하는 중앙 집중식 서비스로 알맞은 분산처리 및 분산 환경을 구성하는 서버 설정을 통합적으로 관리
- **Hive:** 데이터를 모델링하고 프로세싱하는 경우 가장 많이 사용되는 데이터 웨어하우징 용 솔루션(데이터 처리를 위한 배치 처리 구조)

단점

- HDFS에 저장된 데이터를 변경 불가하다.
- 실시간 데이터 분석 같이 신속하게 처리해야 하는 작업에는 부적합하다.
- 너무 많은 버전과 부실한 서포트
- 설정의 어렵다.

구성 요소

Hadoop

Spark

Kafka

클라우드 서비스 공급자별 서비스



> 데이터 웨어하우스 vs 데이터 레이크

01

데이터 웨어하우스

데이터 집합을 대규모로 수집하여 자체 정보에 따라 저장되고 분류 테이블과 데이터 집합은 정형화되고, 데이터는 필요 시 접근할 수 있도록 패키지화됨
데이터를 올바르게 보관하고 필요할 때 호출하려면 모든 데이터를 분석해야 합니다.

단점)

1. 데이터 웨어하우스 시스템에서는 사용자가 특정 테이블에 접근하기 쉬운 반면, 초기 분석과 저장에 시간이 오래 걸리고 리소스가 많이 필요할 수 있음.
2. 게다가 잘못 사용되는 데이터 웨어하우스는 비효율적일 수 있음. 즉각 사용되지 않거나 용도가 분명하지 않은 데이터는 잊히거나 분석에서 제외될 수 있기 때문.
3. 저장 비용이 늘어날 수 있기 때문에, 구조적 이점을 활용하려는 분석가와 IT 전문가는 데이터 웨어하우스의 확장 전략을 신중히 세워야 합니다.

02

데이터 레이크

데이터 웨어하우스가 통제되고 카탈로그화된다면, 데이터 레이크는 모든 데이터가 자유롭게 흐르는 거대한 덩어리. 모든 데이터는 분석 또는 사용 여부와 관계없이, 간헐적으로 사용되더라도 저장. 데이터는 원시 형태로 가져오고 필요할 때만 분석

Hadoop은 하드웨어 측면에서 꽤 경제적이기 때문에 필요 시 손쉽게 확장하여 대량의 데이터를 저장하거나 구문 분석할 수 있으나, 사전 패키지된 테이블과 승인된 데이터 집합을 언제든지 사용할 수 있게 유지하기가 좀 더 어렵게 되므로
데이터 웨어하우스의 이점을 누리기 어려워져 데이터 레이크의 사용이 촉구됨

데이터 레이크에 데이터를 전송할 때는, 데이터의 양과 종류를 늘리는 것에 집중하고 데이터를 사용자가 쓰기 쉽게 가공하는 것은 신경을 덜 쓰게 됨. MR 프레임워크 덕분에 거대 데이터를 가공하는 비용이 줄어들었기 때문에, 데이터의 가공은 사용자에게 맡기고 데이터 레이크는 원본 데이터를 큰 수정 없이 그대로 저장하는 것이 일반적

구성 요소

Hadoop

Spark

Kafka

클라우드 서비스 공급자별 서비스



> Apache Spark

Apache Spark

- 하둡과 마찬가지로 빅데이터 처리 플랫폼/프레임워크. 하둡과 같은 분산형 데이터 컬렉션 상부에서 동작하는 데이터 프로세싱 툴이며, 분산형 스토리지로서의 역할은 수행하지 않음
- Apache Spark는 Apache Kafka가 실시간으로 생성 한 데이터 스트림 (DStream)에 대한 데이터 분석을 수행하는 데 사용. 더 구체적으로 Spark Streaming 모듈이 사용됨

01

Hadoop vs Spark

- **속도**: 스파크의 속도는 MR(하둡)의 속도보다 월등히(100배) 빠름. 하지만, 일반적인 데이터 운영 및 리포팅 상황에서 대부분은 정적인 성향을 띄고, 배치 모드의 프로세싱을 기다릴 수 있다면, 굳이 스파크를 쓰지 않아도 무방하다.
- **Fault tolerance**: 하둡의 경우, 프로세싱 절차마다의 기록을 디스크에 기록하여 failover하는 방식이다. 스파크의 경우, 탄력적 분산형 데이터셋(RDD)을 활용하여, 메모리 내 또는 디스크에 저장하여 완벽 복구할 수 있는 탄력성 보장
- **신뢰성**: 하둡은 분산형 플랫폼이기 때문에 고장에 덜 취약해 기본 데이터를 항상 이용할 수 있고, 상시 서비스 역량이 요구되는 웹 기업들이 이 DB를 선택함
- **비용**: 둘 다 오픈소스이므로 무료(추가 자원 비용 제외)
- **보편성**: 스파크는 MySQL, S3, HDFS 등 모든 곳에서 데이터를 불러올 수 있음
- **사용 사례**
 - 하둡**: 많은 데이터를 저장할 수 있으므로 전체 모니터링 및 소매 기업을 위한 추천 엔진, 보안 및 위험 관리 등에 활용
 - 스파크**: 실시간 상호작용 DA를 통한 더욱 광범위한 개인화 제공 기능, 소매 기업 및 IoT 기업들이 활용 + 머신러닝

02

두 개는 경쟁적인 대상인가? 아니오

- 실시간 데이터에 대한 상호작용이 즉각적으로 필요한 게 아니고 분산처리가 필요하다면 하둡, 실시간 처리가 필요하고 머신러닝 엔지니어링에는 스파크가 조금 더 적합
- 최근에는 **Hadoop + Spark의 연계**가 하나의 공식이 되었다. 하둡의 YARN 위에 스파크를 얹고, 실시간성이 필요한 데이터는 스파크로 처리하는 방식으로 아키텍처를 구성하여 동작.
- Spark는 MapReduce를 대체. Spark 상의 데이터 처리는 스크립트 언어를 사용할 수 있다 (자바, 스칼라, 파이썬, R 등)

구성 요소

Hadoop

Spark

Kafka

클라우드 서비스 공급자별 서비스



> Apache Kafka

Apache Kafka

- 빠르고 확장 가능한 작업을 위해 데이터의 분산 스트리밍, 파이프 라이닝 및 재생을 위한 실시간 스트리밍 데이터를 처리하기 위한 목적으로 설계된 오픈 소스 분산형 게시-구독 메시징 플랫폼. 서버 클러스터 내에서 데이터 스트림을 레코드로 유지하는 방식으로 작동하는 브로커 기반 솔루션
- 많은 양의 데이터를 안정적으로 보관하고 처리할 수 있음(Spark와 유사하게 대용량 스트림 처리에 사실상의 표준)

+ Apache?

Apache

- 아파치: 세계에서 가장 많이 쓰는 웹 서버중 하나이다.
- 아파치는 Apache재단에서 만든 HTTP서버이며 이 서버가 굉장히 다양하고 기능적인 면에서 우수. 또 구축이 쉽다는 이유 때문에 많이 사용하고, 대부분의 중소기업들은 무료이기 때문에 많이 사용



등등...



구성 요소

Hadoop

Spark

Kafka

ELK/EFK

클라우드 서비스

공급자별 서비스



> 클라우드에서 각 도구를 사용할 수 있는 서비스

Cloud Services

- 단순히 서버(EC2)에 직접 sw를 설치할 수도 있지만, 대부분의 클라우드는 **완전관리형 솔루션**을 제공함

완전관리형 솔루션: AWS는 완전관리형 서비스는 AWS에서 모두 관리해주고, 사용자는 설정만으로 또는 설정하지 않아도 쉽게 서비스 설치, 백업, 가용성 등에 대한 부분을 보장 받을 수 있음

	AWS	GCP	Azure
Kafka	MSK, Kinesis(Kafka Broker)	Dataflow	HDInsight
Hadoop	EMR	Dataproc	HDInsight
Spark	EMR	Dataproc	HDInsight
데이터 웨어하우스	Redshift	BigQuery	Synapse
데이터레이크	S3	BigLake, Storage	Data Lake

구성 요소

Hadoop

Spark

Kafka

ELK/EFK

클라우드 서비스

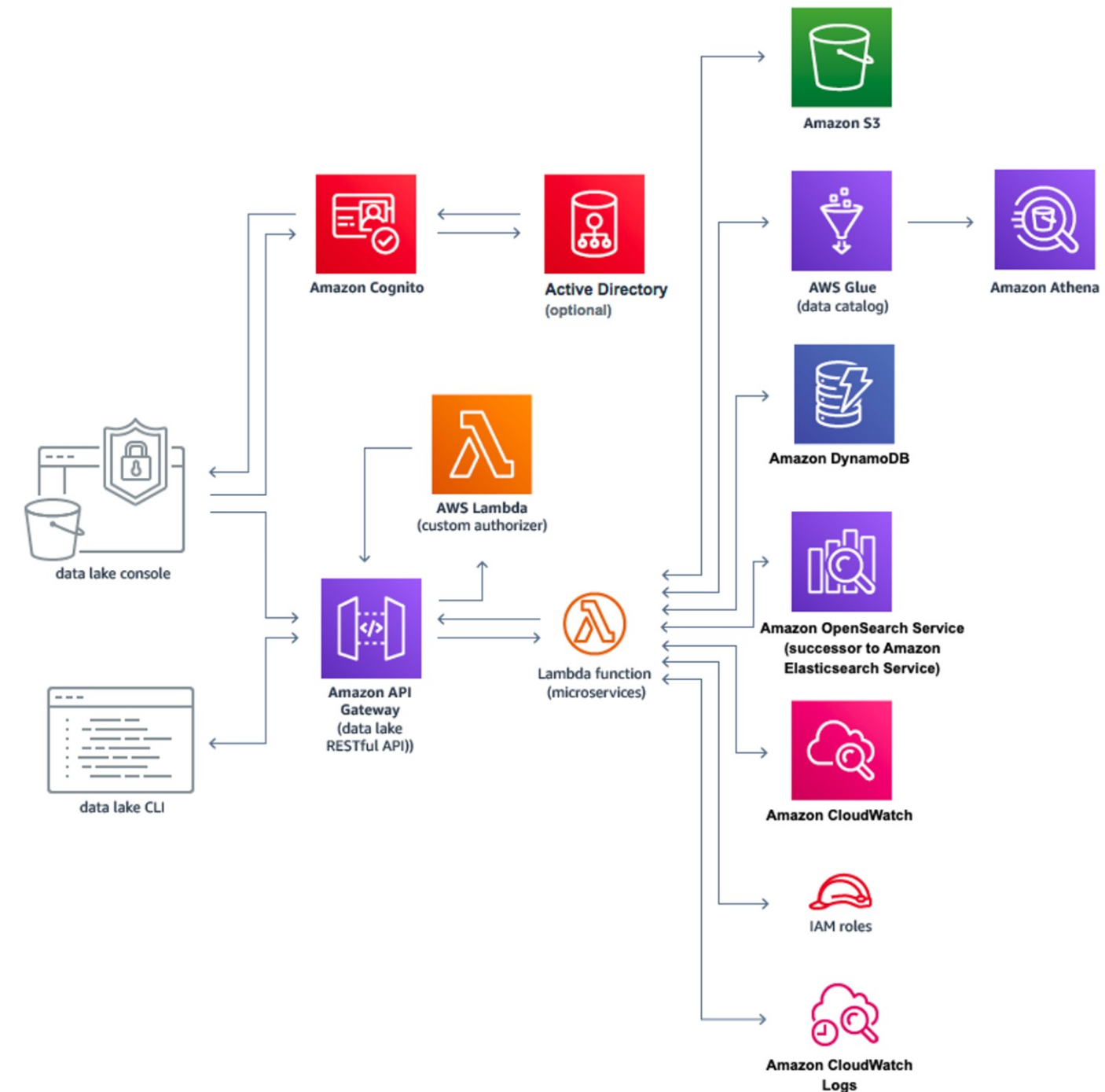
공급자별 서비스



> 클라우드에서 각 도구를 사용할 수 있는 서비스

AWS Data lake architecture

- AWS 데이터 레이크 아키텍처 링크(<https://aws.amazon.com/ko/solutions/implementations/data-lake-solution/>)



구성 요소

Hadoop

Spark

Kafka

ELK/EFK

클라우드 서비스

공급자별 서비스



> 멀티 클라우드? 하이브리드 클라우드?

멀티 클라우드

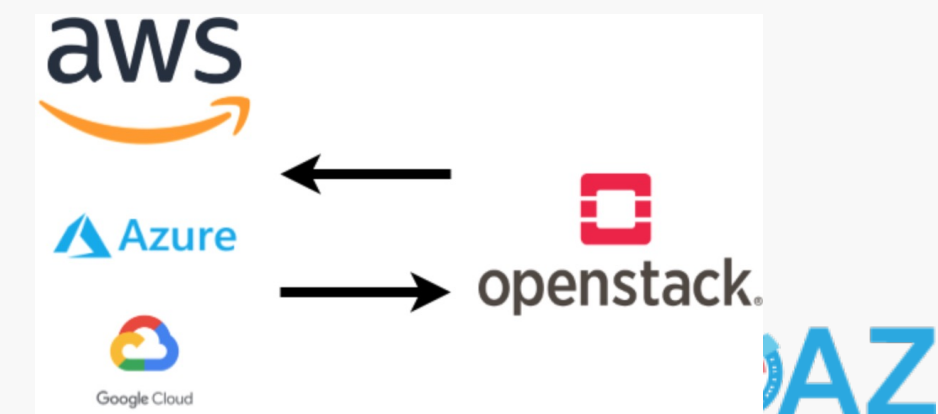
- 멀티클라우드는 2곳 이상의 클라우드 벤더가 제공하는 2개 이상의 퍼블릭 또는 프라이빗 클라우드로 구성된 클라우드 접근 방식(왼쪽 그림)
- 클라우드 사이의 워크로드 이식성, 상호연결, 오케스트레이션, 통합관리 없이 퍼블릭/프라이빗 클라우드를 개별적으로 사용

하이브리드 클라우드

- 하이브리드 클라우드 모델은 온프레미스 IT(기존 인프라 및 프라이빗 클라우드)를 Google Cloud Platform(GCP), Amazon Web Services(AWS) 또는 Microsoft Azure와 같은 퍼블릭 클라우드와 클라우드 서비스 공급업체(CSP)의 오프프레미스 리소스 또는 서비스와 결합(오른쪽 그림)
더 엄격하게 정의하면 하이브리드 클라우드는 프라이빗 및 퍼블릭 클라우드와 CSP를 모두 포함할 수 있는 다양한 클라우드의
- 조합으로 구축된 서비스입니다. 3-티어 애플리케이션 스택에서 프레젠테이션 서비스는 퍼블릭 클라우드에, 애플리케이션 서비스는 관리형 프라이빗 클라우드에, 데이터베이스 서비스는 온프레미스에 있을 수 있습니다.
- 클라우드 사이의 이식성, 상호연결, 오케스트레이션, 통합관리를 통해 퍼블릭/프라이빗 클라우드를 상호 운용적으로 사용

차이점

- 멀티 클라우드는 여러 벤더가 제공하는 동일한 유형(퍼블릭 또는 프라이빗)의 클라우드를 2개 이상 배포하는 것을 말하며, 하이브리드 클라우드는 여러 배포 유형이 있고 이들 사이에 통합이나 오케스트레이션이 특정 방식으로 이루어지는 것



그 외



> 그 외 사용 가능한 서비스들

Airflow: 작업 흐름 관리

- 데이터 파이프라인이 복잡해지만 각 데이터 처리 프로세스 또한 잘 구조화하여 관리해야한다. 특히 job간의 의존성이 있는 경우 앞의 작업에 끝난 후 뒤의 작업을 실행하는 것이 중요하다. Airflow는 이런 작업들을 workflow 도구를 이용하여 자동화하고 쉽게 스케줄링할 수 있으며, 각 task의 의존성에 따라 잘 실행시켜준다.

Ambari: 클러스터 관리

- 클러스터가 커짐에 따라서 여러가지 솔루션을 직접 여러 머신에 설치하여 ssh로 접속하여 설정을 변경하고 관리하기 어렵다. Ambari는 클러스터에 설치된 여러 솔루션들의 설정값을 관리하고, 각 요소들을 중지, 시작하는 것을 웹 인터페이스를 통해 할 수 있다.

NiFi: 데이터 흐름을 다룸

- 데이터를 처리하는 과정이 복잡해지고 실시간 처리 요구가 증가함에 따라 작은 수준의 처리를 위한 코딩을 하는 데 시간이 걸린다.(ex Kafka에 들어온 메시지에 특정 필드를 추가하는 작업). 간단한 작업이지만 많은 시간이 요구되는 작업에 대해서 이를 개발하기 위해 코딩하는 시간과 모니터링까지 하기엔 관리 차원에서 고민이 많아질 수 밖에 없다. NiFi는 여러가지 작업에 대한 프로세서가 준비되어있어 빠르게 끝낼 수 있고, 프로세서 사이의 데이터 흐름을 쉽게 모니터링할 수 있으며 수평적 확장(노드 추가)이 가능

Druid: 빠른 집계

- 빅데이터 분석을 통해 알고 싶은 내용은 집계(group by)의 결과인 경우가 많은데, 이를 위해 매우 많은 계산이 요구된다. Spark를 이용하면 빠를 수 있으나, 실시간 대용량 데이터를 위해서는 아주 큰 메모리가 탑재된 Spark 클러스터가 필요하다. Druid는 데이터를 미리 만들어 다차원에 대한 집계 결과나 top N 질의를 실시간으로 응답해줌.

Zepplin: 쉽게 대시보드 생성

- 데이터 처리와 정제를 아무리 거쳐도 사용자에게 이를 제공하지 않으면 무용지물이다. 여러 가지 데이터를 정리하여 보고서나 대시보드 같은 것을 제공할 때 제플린을 사용한다. 주피터와 비슷한데, 코드 실행 결과보다 여러 가지 시스템에 대한 질의 결과를 보여주는 데 좀 더 초점이 맞춰져 있다.

실습

실습: 기업들의 데이터 파이프라인 분석.

I 데이터 파이프라인 예시

User performs "Transformation" in the UI with new, which EC2 handles Data Transformation.

Save Query Result to DynamoDB for those don't need refreshes to feed faster to D3

"Load" to S3 directly from EC2

"Extract" & "Transform" & "Load"

Amazon EC2

EC2 Handles:

- API Data Pulls
- Custom Uploads

Modifies inside EC2 and send it to S3 Directly

{ REST }

Lambda hits 3rd Party APIs to extract data.

User can directly hit API Gateway to:

- Backfill Data
- Any special circumstances

AWS Lambda

Amazon Kinesis

Amazon S3

S3 Stores all raw data in its new bucket per source, and partition by selected keys by user

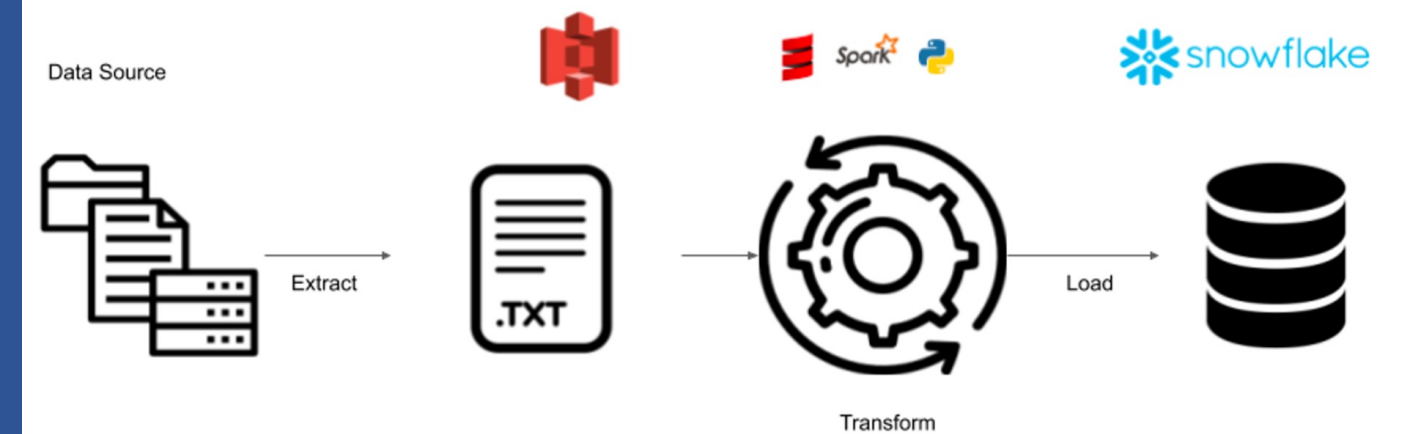
Amazon DynamoDB

Amazon Athena

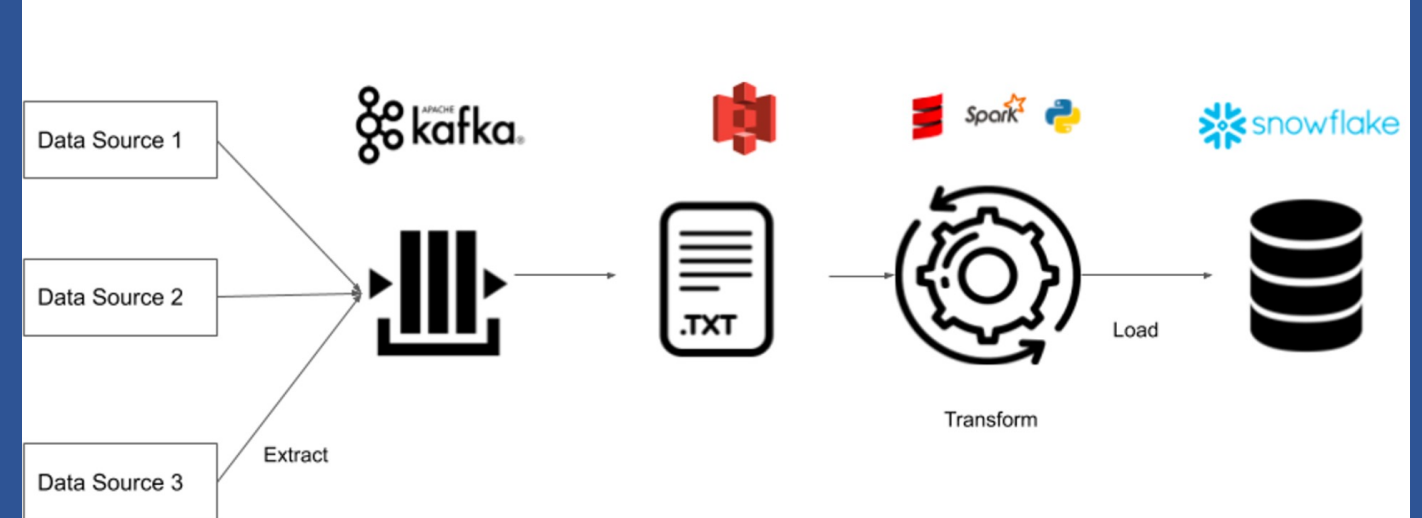
FRONT-END

Through UI, user can transform existing data and load it as new data form

Batch ETL



Streaming ETL

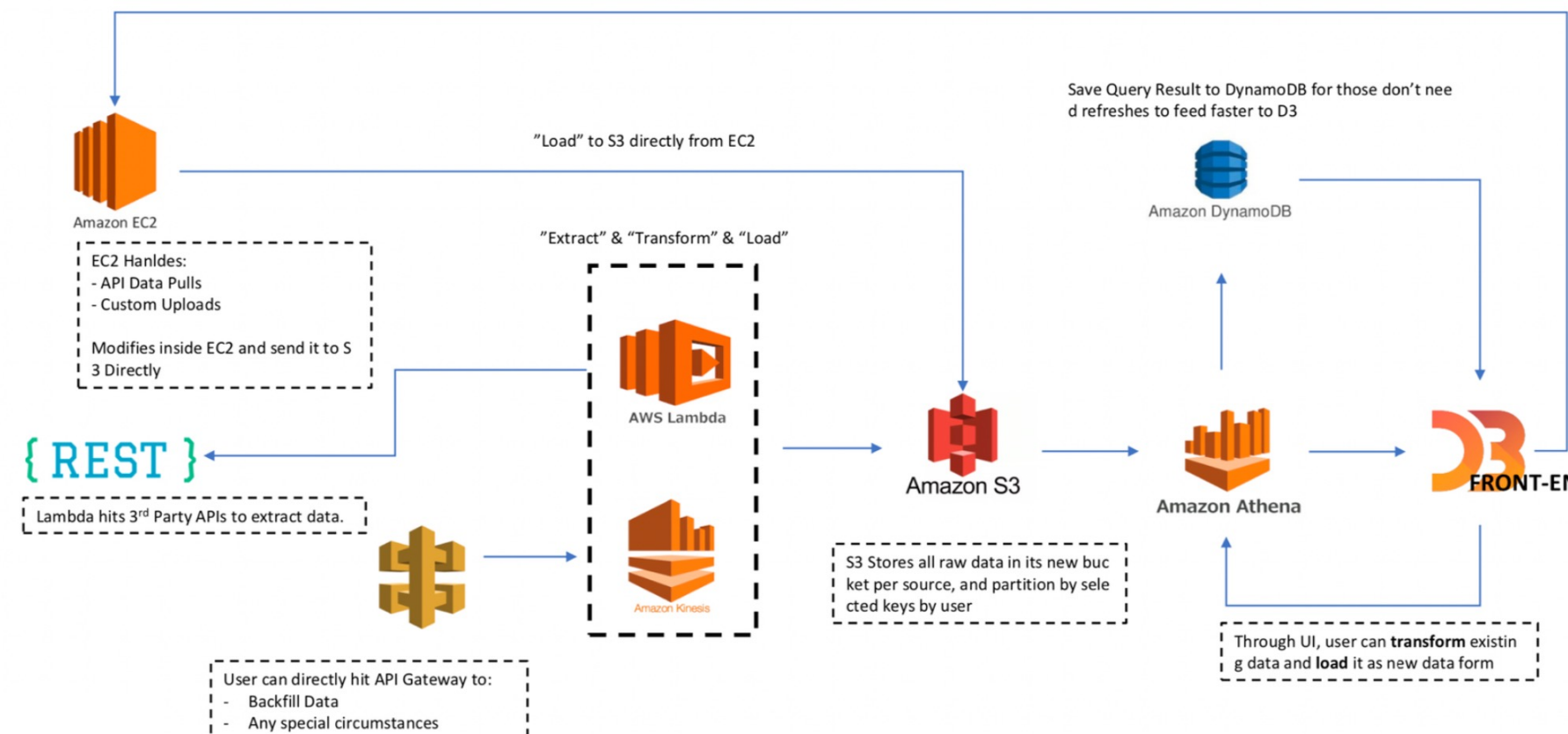


실습: 기업들의 데이터 파이프라인 분석.

팀 당 셋 중 하나의 파이프라인을 골라서 분석해보기!

I 데이터 파이프라인 예시

User performs "Transformation" in the UI with new, which EC2 handles Data Transformation.



> 데이터 원천이 무엇인지

어떤 특징을 가진 데이터에 대한 파이프라인인지

> 목적이 무엇인지

최종 사용자가 누구이며, 어떤 용도로 이용할 데이터에 대한 파이프라인인지

> ETL? ELT? 데이터 웨어하우스? 데이터 레이크?

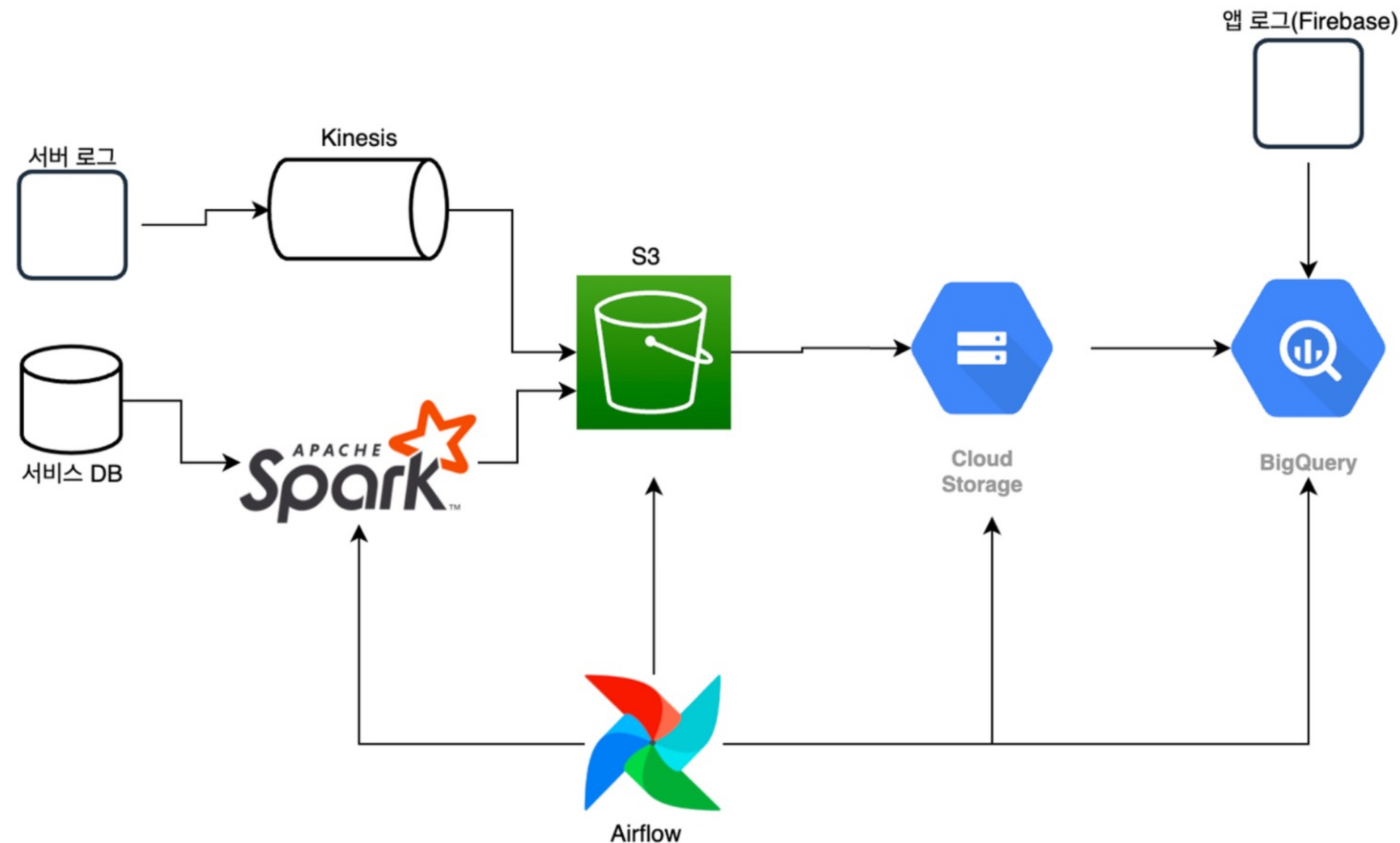
> 각 서비스들의 역할

클라우드 서비스보다는 데이터 처리 서비스를 중심으로. 멀티? 하이브리드?

> 개선점? 의문점?

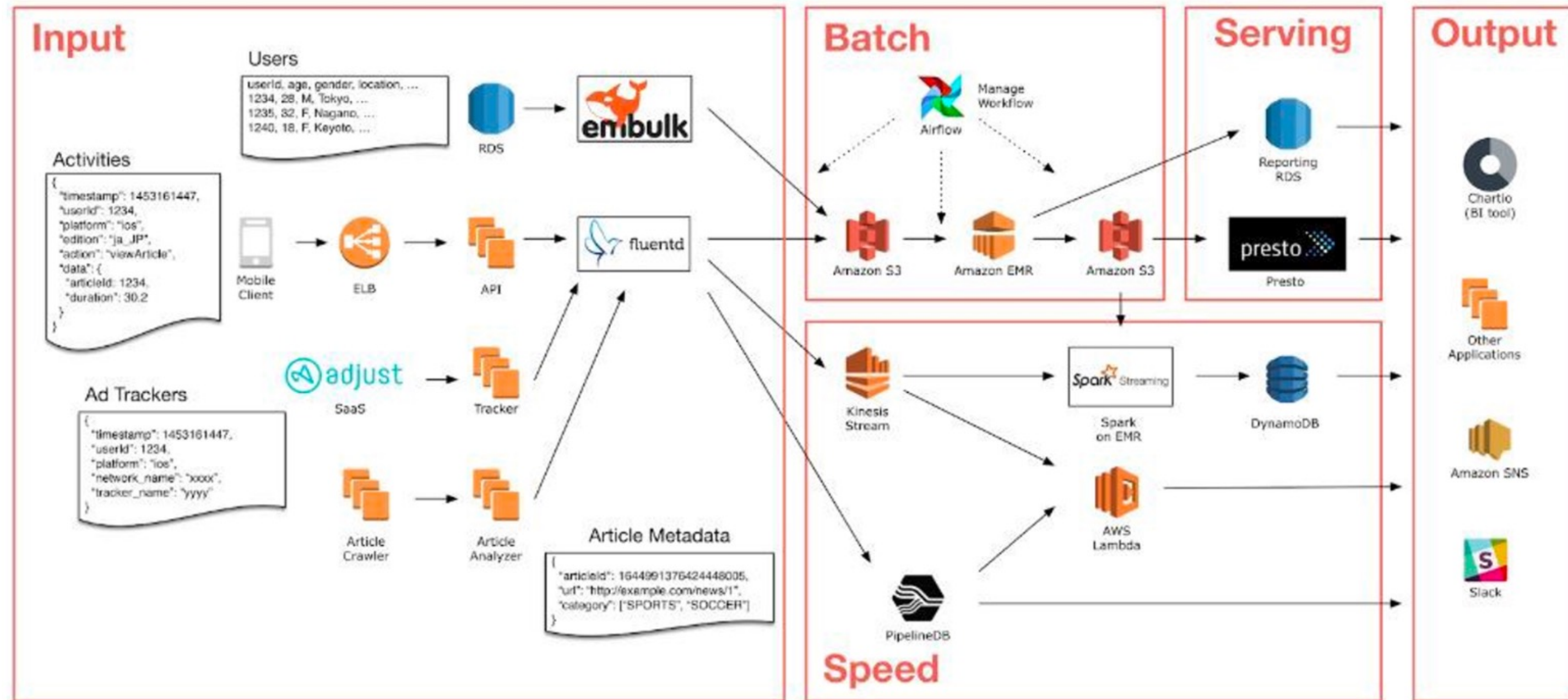
7000만 학생이 가입한 인공지능 AI 수학 공부앱, [관다].

출처: 관다 팀블로그(<https://blog.mathpresso.com/tagged/data-infrastructure>)



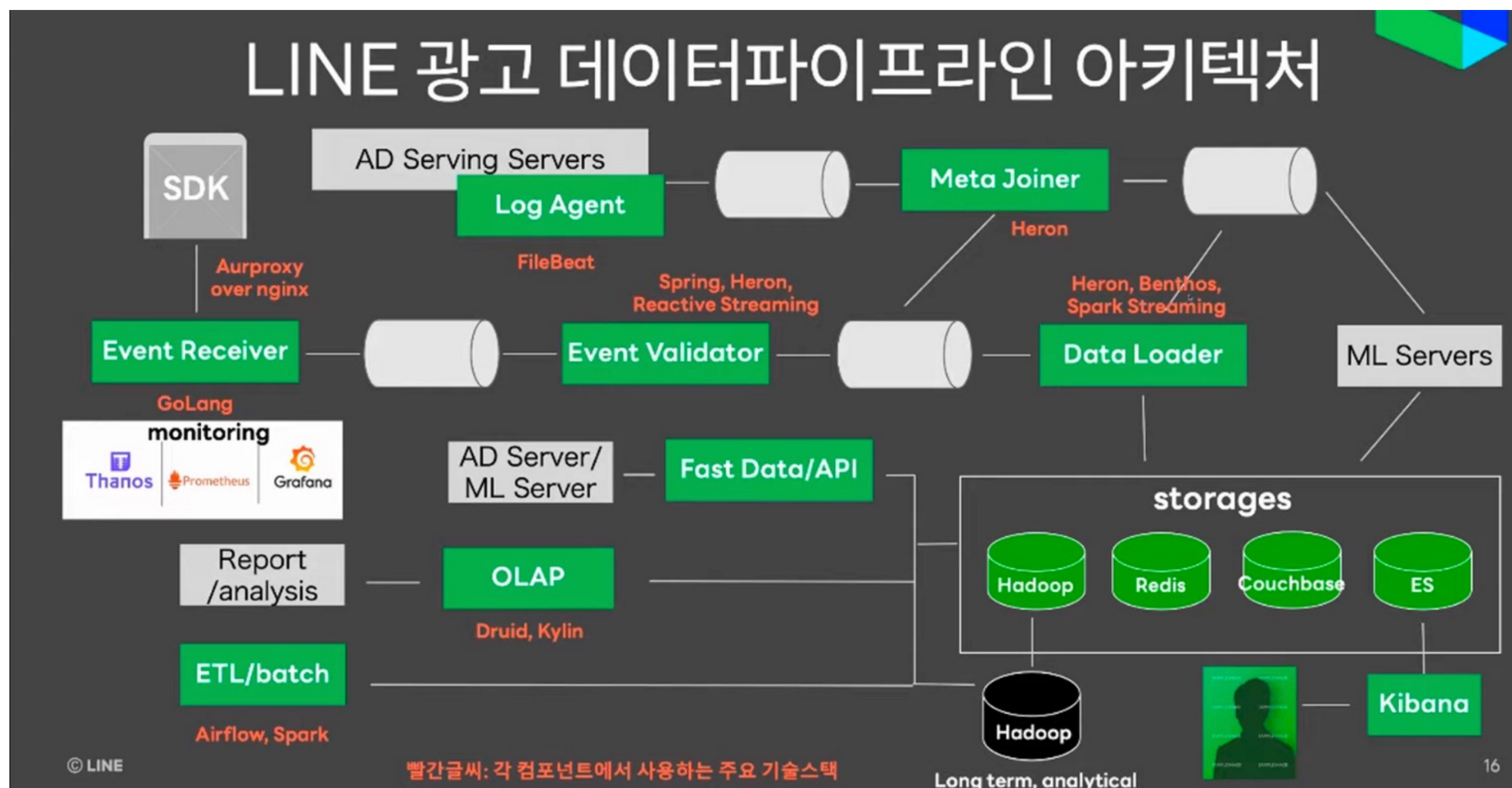
여행의 모든 것, 한번에 쉽게 [야놀자].

출처: 야놀자 발표 자료(https://docs.google.com/presentation/d/11C_BKio0DZlop_ZjJk7ogxQtWV5qHlr-hHjw277z64k/edit#slide=id.g586daab0bd_0_366) 37페이지 참고



전 세계 1억 8천만 명 이상의 사용자, 글로벌 광고 플랫폼 [LINE Ads].

출처: 라인개발실록 유튜브(<https://www.youtube.com/watch?v=rCbzilpjsdY&t=3s>) 8:40초 참고



실제 파이프라인 구성[•] 관련 자료

실제 구성 참고 자료

➤ [티아카데미 빅데이터 파이프라인 구성 실습 1, 2, 3강]

https://www.youtube.com/watch?v=0xwM_PG_DEc&list=PL9mhQYIIEhfgzvxjzWCRYJ80yeyei50

https://www.youtube.com/watch?v=bBjuptUpQ_M&list=PL9mhQYIIEhfgzvxjzWCRYJ80yeyei50&index=2

https://www.youtube.com/watch?v=_H9_k3Jh0zQ&list=PL9mhQYIIEhfgzvxjzWCRYJ80yeyei50&index=3

➤ [데이터 파이프라인 자동화 실습(AWS S3, Lambda 등)]

https://heung-bae-lee.github.io/2020/03/01/data_engineering_09/


참고문헌

Document



빅 데이터와 Hadoop이 자주 함께 거론되는 이유

Hadoop과 빅 데이터는 밀접하게 관련되어 있어서 함께 거론되거나, 최소한 같이 등장하는 경우가 많습니다. 빅 데이터는 그 의미가 아주 넓어 거의 모든 것과 연관될 수 있습니다. 빅 데이터는 오늘날 디지털 세상에서 즐겨야 할 한 분야로 급부상하고 있고, Hadoop은 빅 데이터 내에서 답을

 <https://www.tableau.com/ko-kr/learn/articles/big-data-hadoop-explained>

하둡(Hadoop)

하둡(Hadoop)이란? 하나의 성능 좋은 컴퓨터를 이용하여 데이터를 처리하는 대신 적당한 성능의 범용 컴퓨터 여러 대를 클러스터화하고 큰 크기의 데이터를 클러스터에서 병렬로 동

 <https://velog.io/@ha0kim/2021-03-02>



data engineering (데이터 웨어하우스 vs 데이터 레이크)

데이터 웨어하우스 vs 데이터 레이크 데이터 레이크라는 개념은 비교적 최신의 개념이다. 데이터 웨어하우스라고 하는 MySQL, PostgreSQL 같은 RDBMS 프로그램들을 넘어서 데

https://heung-bae-lee.github.io/2020/02/22/data_engineering_07/

구분	데이터 레이크	데이터 웨어하우스
Data Structure	Raw	Processed
Purpose of Data	Not Yet Determined	In Use
Users	Data Scientists	Business Professionals
Accessibility	High / Quick to update	Complicated / Costly

데이터 웨어하우스란 무엇입니까? | 주요 개념 | Amazon Web Services

데이터 웨어하우스는 보다 정보에 입각한 의사 결정을 내릴 수 있도록 분석 가능한 정보의 중앙 리포지토리입니다. 데이터는 트랜잭션 시스템, 관계형 데이터베이스 및 기타 소스로부터

 <https://aws.amazon.com/ko/data-warehouse/>



ETL과 ELT 정리

 <https://sda1547.gitbook.io/data-engineering/database/etl-elt>

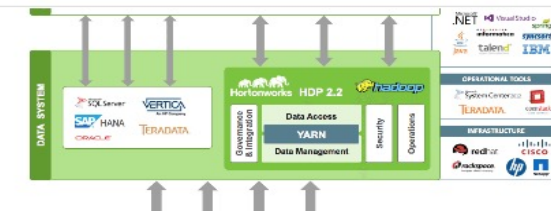
1.0.0

ETL과 ELT 정리

빅 데이터 프레임워크, 솔루션들의 목적과 역할

여러 프레임워크와 솔루션의 목적과 역할을 알아봅니다 | 데이터 엔지니어링을 시작하면서 굉장히 많은 프레임워크와 솔루션들을 접했다. Hadoop과 같이 사실상 표준으로 쓰이는 것

 <https://brunch.co.kr/@toughprogrammer/24>



19기 Engineering Base

감사합니다 •

데이터 파이프라인 실습

18기 엔지 금나연