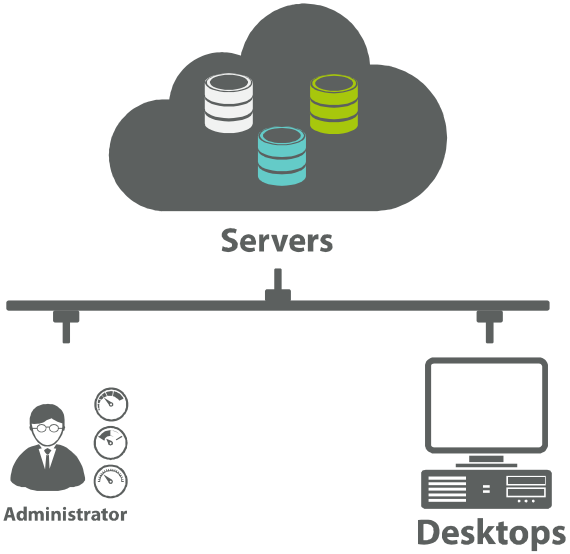
클라우드 기반 데이터레이크 및 분석

Data Lake는 어떻게 구축해야 하는가?

MEGAZONE CLOUD



Index

Data Lake 구현 방식

Data Lake 구현 단계

기획단계

- 구축방식선정
- On-Premise vs Cloud
- 구축 로드맵 수립

설계단계

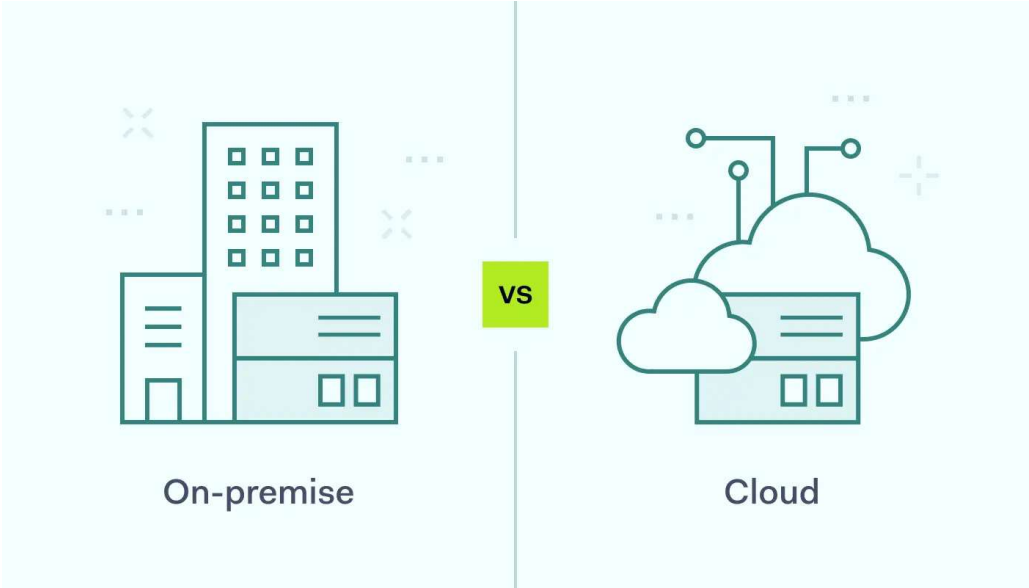
- 참조 모델(카파 vs 람다) 기반 Layer 구분
- Layer별 구성요소 정의
- 구성요소별 필요 기능 정의
- 자체개발 vs 솔루션 도입 의사결정
- H/W 아키텍처 설계

구현단계

- 단계별 서비스 목표 설정
- 세부 서비스 단위 구현 및 가시화

윤선웅, *차세대 빅데이터 플랫폼 Data Lake*, 좋은땅, 2021, p.64.

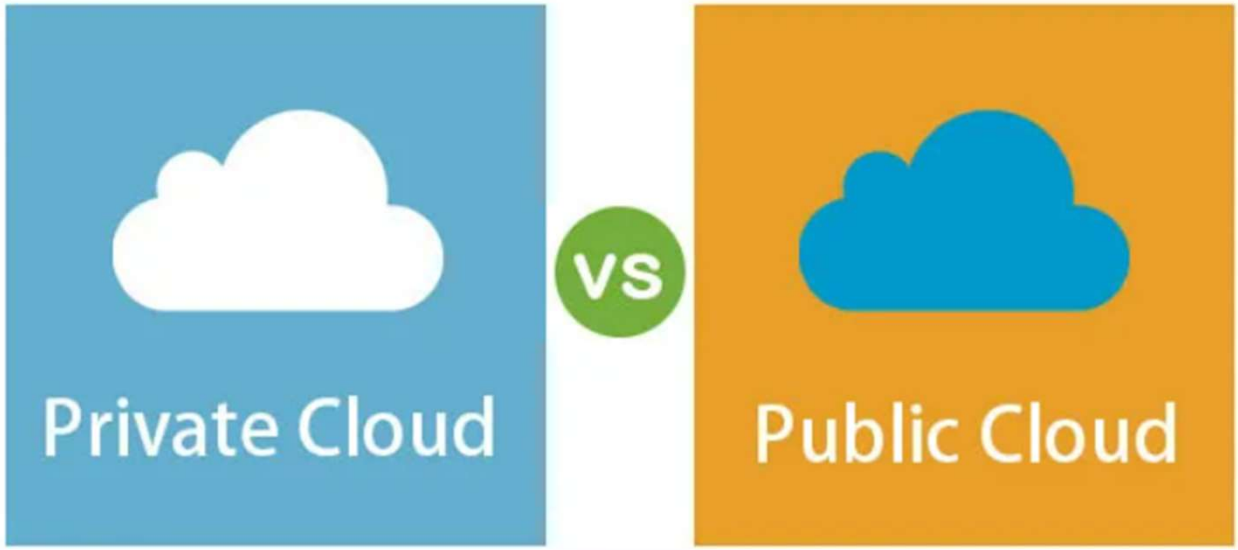
On-Premise vs. Cloud



- 리소스 소요 변동 폭이 적을 시 유리
- 실시간 처리 비중이 높을 시 유리
- 향후 확장 가능성 적을 경우 유리
- 전사 공통 비용 기반으로 운영 시 유리
- 자사의 데이터 센터에 모두 구성

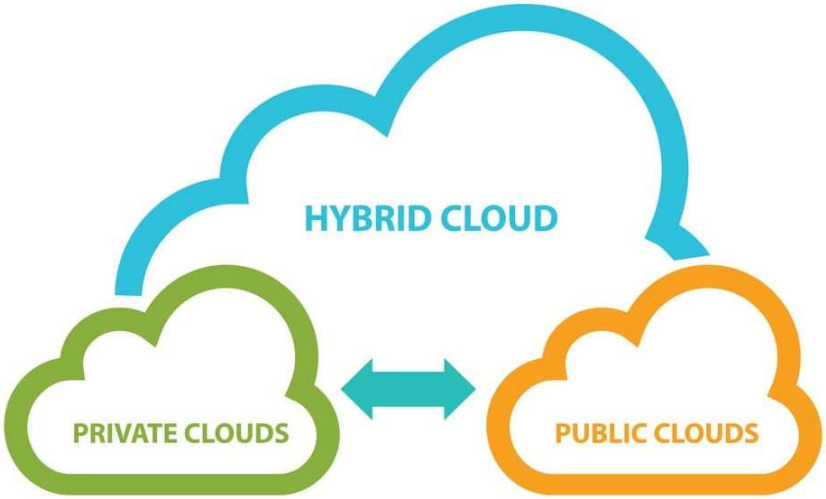
- 리소스 소요 변동 폭이 클 경우 유리
- 배치성 업무 처리 비중이 높을 시 유리
- 향후 확장 가능성 높을 경우 유리
- 조직별 사용량에 따른 비용 배분 시 유리
- private, public, hybrid

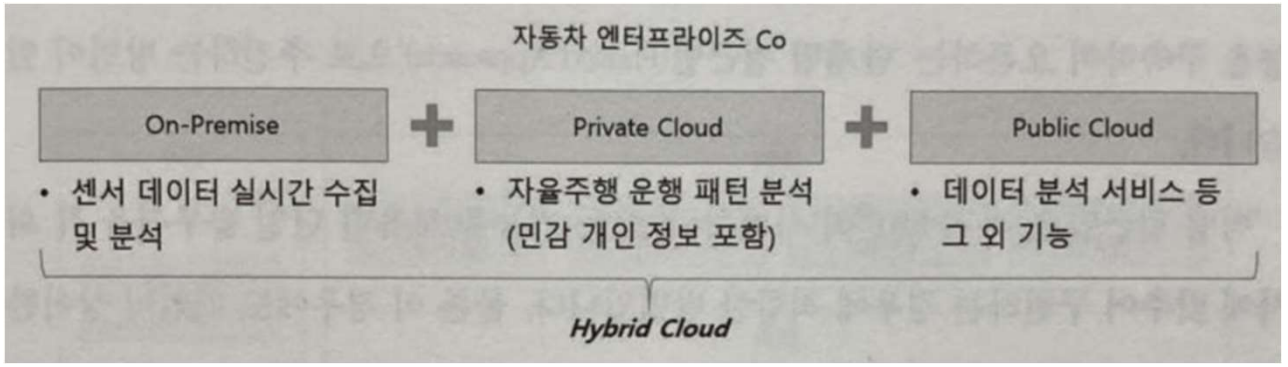
Private Cloud vs. Public Cloud



Private Cloud	Public Cloud
• 엄격한 데이터 보안 규제/정책 존재	• 민감 데이터 보안, 규제 이슈 적을 시
• 구성원의 충분한 구현 역량/비용 보유	• 자사의 IT 구현 역량 충분하지 못할 시
• 장기간에 단계별로 기능 구현 시	• 단기간에 적은 비용으로 구현 필요 시
• Cloud 서비스의 자사 핵심 역량화	• 단순히 Cloud 서비스를 이용
• 필요한 서비스가 외부 벤더에 없을 시	• 필요한 서비스를 외부에서 조달 가능 시
• 급격한 리소스 수요 변동 적을 시	• 급격한 리소스 수요 변동 존재

Hybrid Cloud

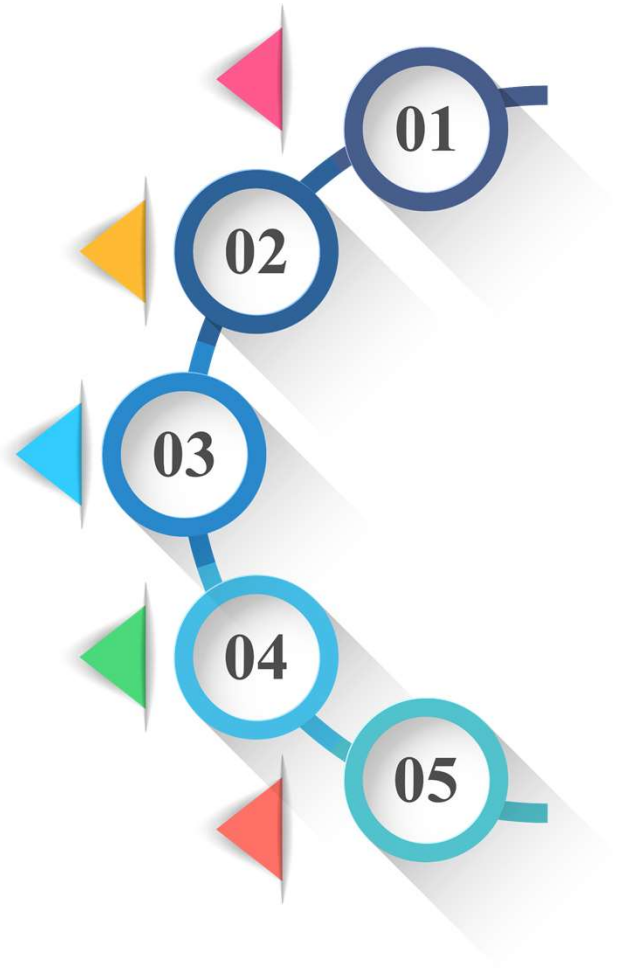




윤선웅, *차세대 빅데이터 플랫폼* ***Data Lake***, 좋은땅, 2021, p.69.

Data Lake 구축 로드맵

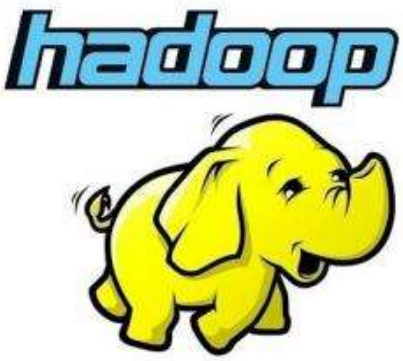
Data Lake 구축 로드맵



- Data Lake 구축은 장기간 소요되는 전사적 전환 과제
- 빅뱅 접근법(Big Bang Approach)
  - 모든 기능을 보유한 단일 솔루션을 각 회사에 맞추어 구현
- 단계별 접근법(Phased Approach)
  - 단계별로 기능을 구축하여 오픈

https://www.pngwing.com/en/free-png-bmwak

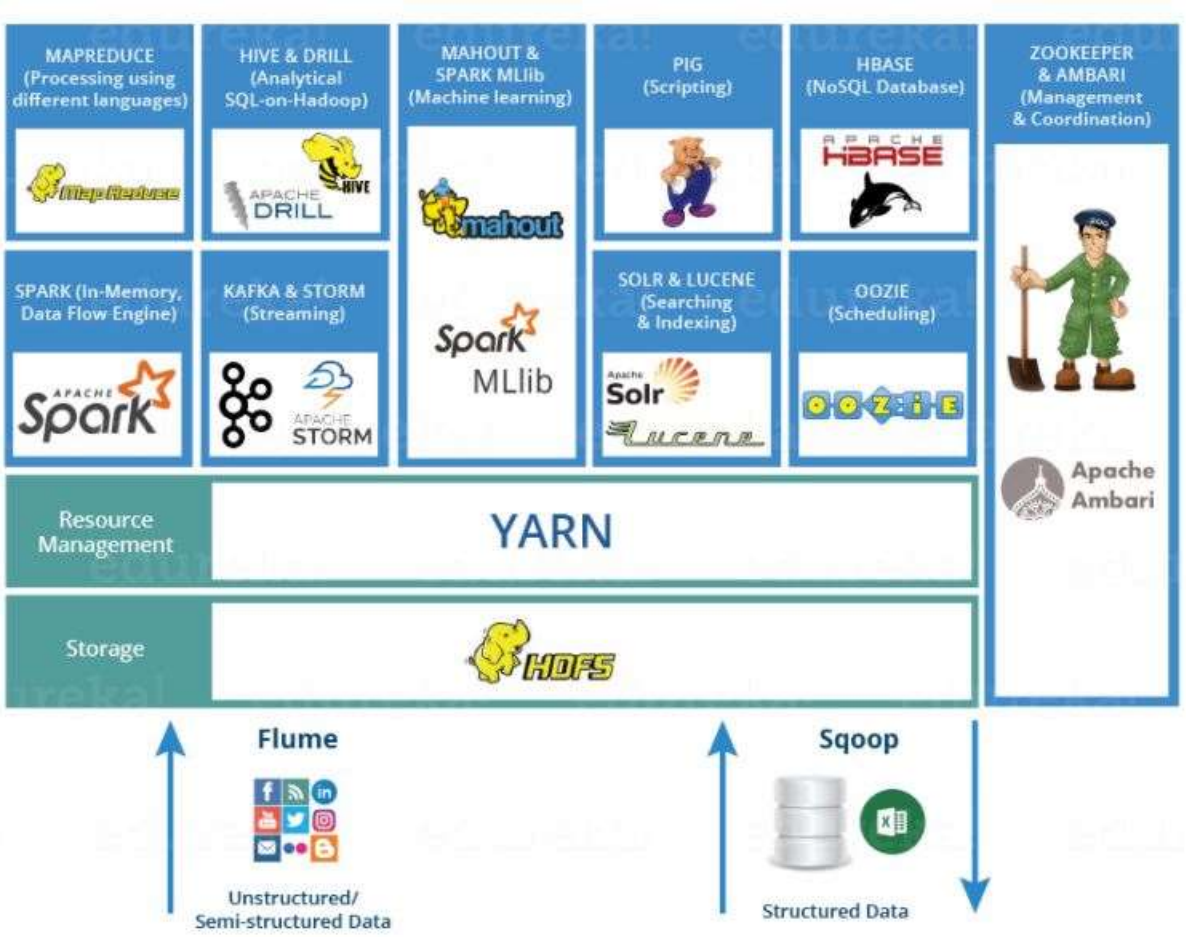
Hadoop 기반 솔루션 vs. Cloud 솔루션



- Data Lake가 표방하는 특징들을 수용 가능한 현식적으로 거의 유일한 솔루션
- 대부분의 기업에서 솔루션으로 채택 중
- 빅데이터 분산 처리를 위한 오픈소스 기반 솔루션
- Data Lake를 구성하는 영역별 다양한 오픈소스 솔루션 제공

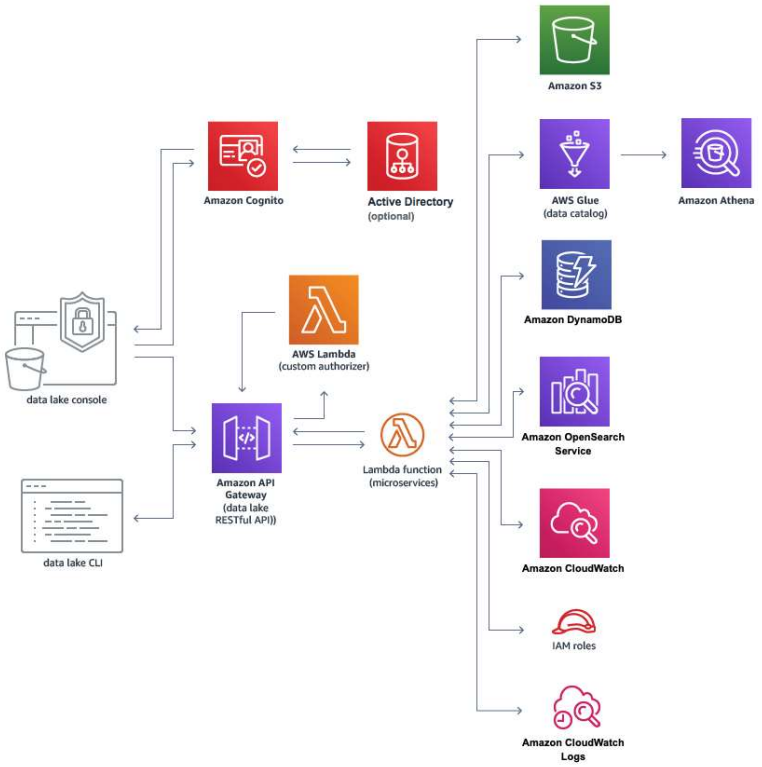
- 각 CSP별로 클라우드에서 서비스 형태로 제공

Hadoop 기반 솔루션 vs. Cloud 솔루션

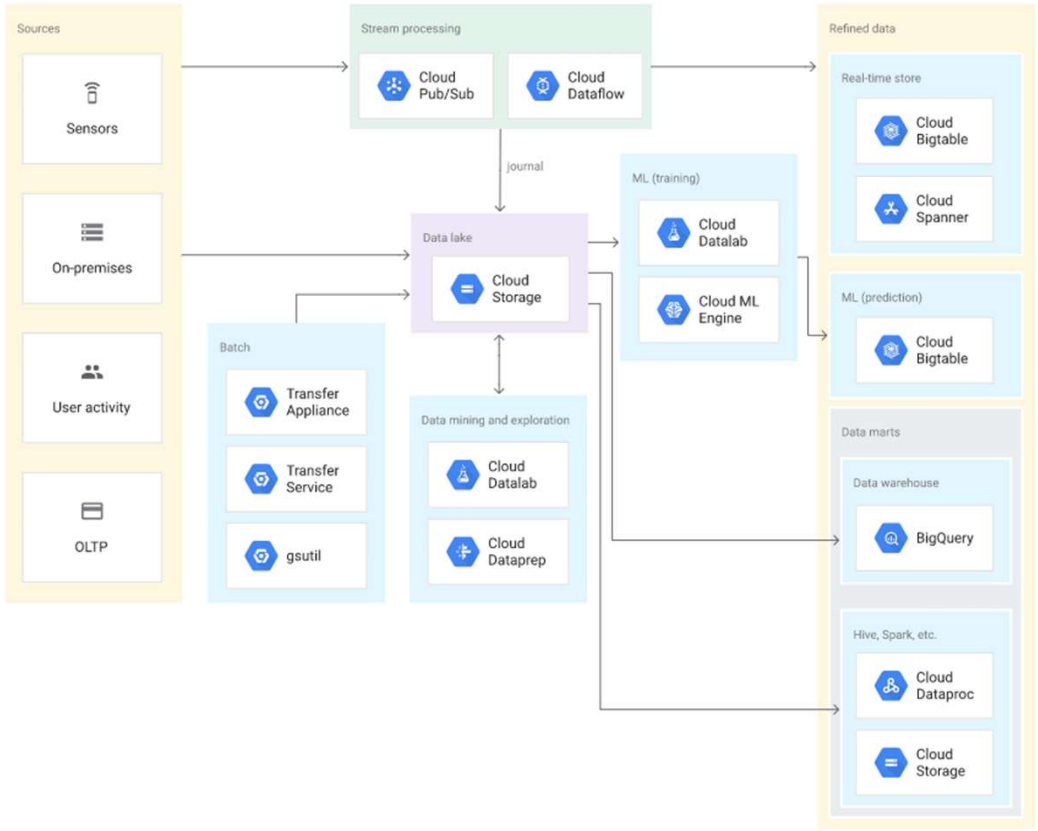


https://velog.io/@yunaaa/Hadoop-Ecosystem

Hadoop 기반 솔루션 vs. Cloud 솔루션 (Cont.)



https://aws.amazon.com/ko/solutions/implementations/data-lake-solution/



https://cloud.google.com/architecture/build-a-data-lake-on-gcp?hl=ko

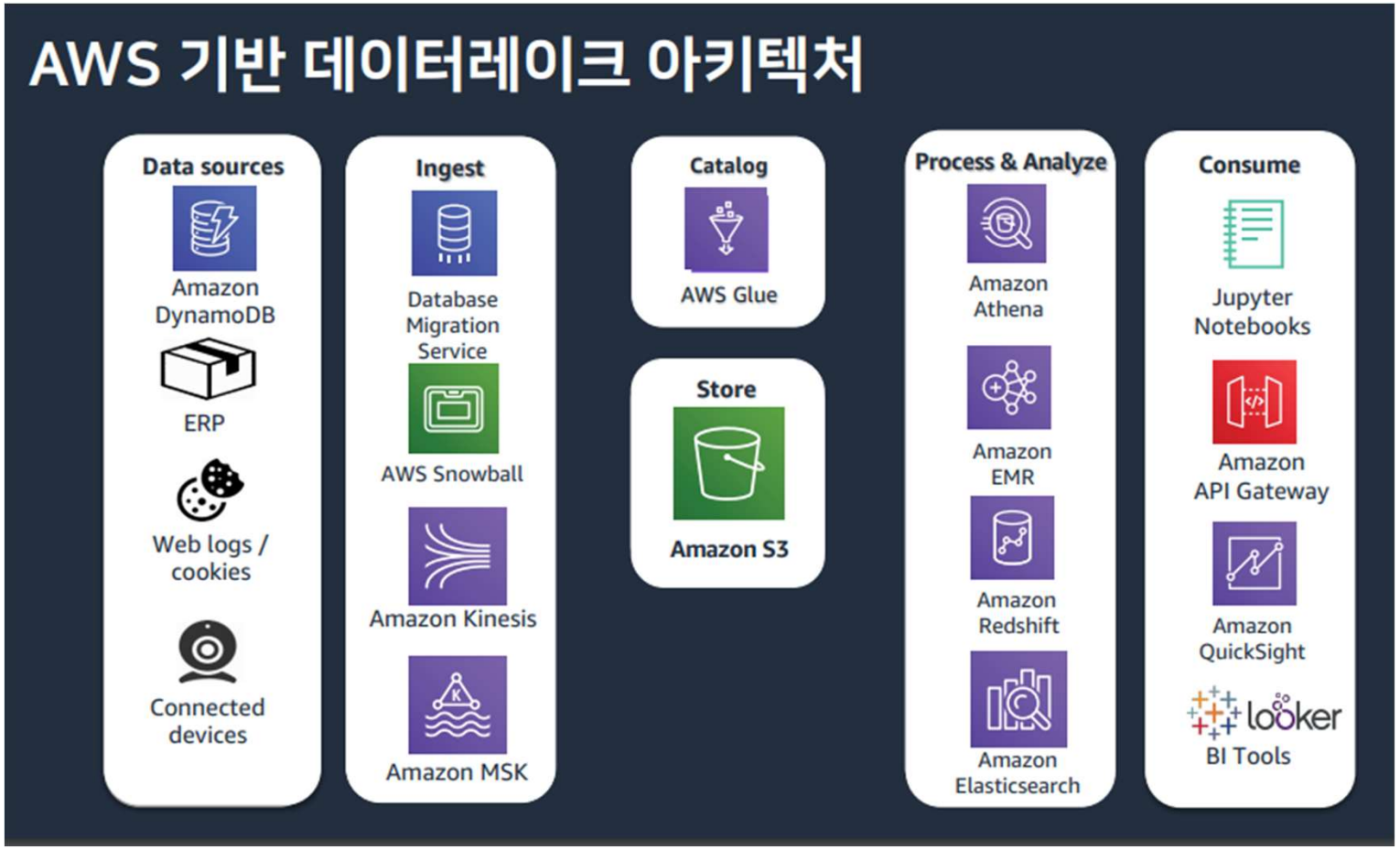
Hadoop 기반 솔루션 vs. Cloud 솔루션 (Cont.)

	Hadoop Solution	Cloud Solution
장점	• Open-Source 기반으로 솔루션 구매 비용 절감(상용 솔루션 구매 시에는 일부 증가) • 각 기업에 최적화된 형태로 커스터마이즈 • 구성원의 Hadoop에 대한 전문 역량 확보 • 특정 벤더에 Lock-In 가능성 감소	• Lake 구축 위한 기간, 인력, 인프라 초기 투자 최소화 • 다양한 기술 요소별 전문 인력 확보 불필요 • 유지보수/운영 부담 최소화(솔루션 벤더에서 기술적 이슈 해결)
단점	• Lake 구축에 많은 기간, 인력, 인프라 초기 투자 필요 • 다양한 기술요소별 전문 개발/운영 인력 소싱 어려움 • 유지보수/운영 부담 증가(상용 솔루션 구매 시 일부 완화 가능)	• 솔루션 이용량에 따른 이용 요금 부과(지속적 많은 활용 기업 비용 부담 증가) • 각 솔루션의 기능을 해당 기업에 최적화 되도록 커스터마이즈 불가) • 솔루션 벤더에 대한 의존도 높아 Lock-In 가능성 증가하고 구성원 역량 증대 제한

윤선웅, *차세대 빅데이터 플랫폼 Data Lake*, 좋은땅, 2021, p.74.

Building Data Lake on AWS

AWS 기반 데이터레이크 아키텍처



https://youtu.be/eQjkwhyOOmI

Data Lake Pipeline in AWS

Data Lake Pipeline in AWS (Cont.)

Apache Spark 소개

Apache Spark

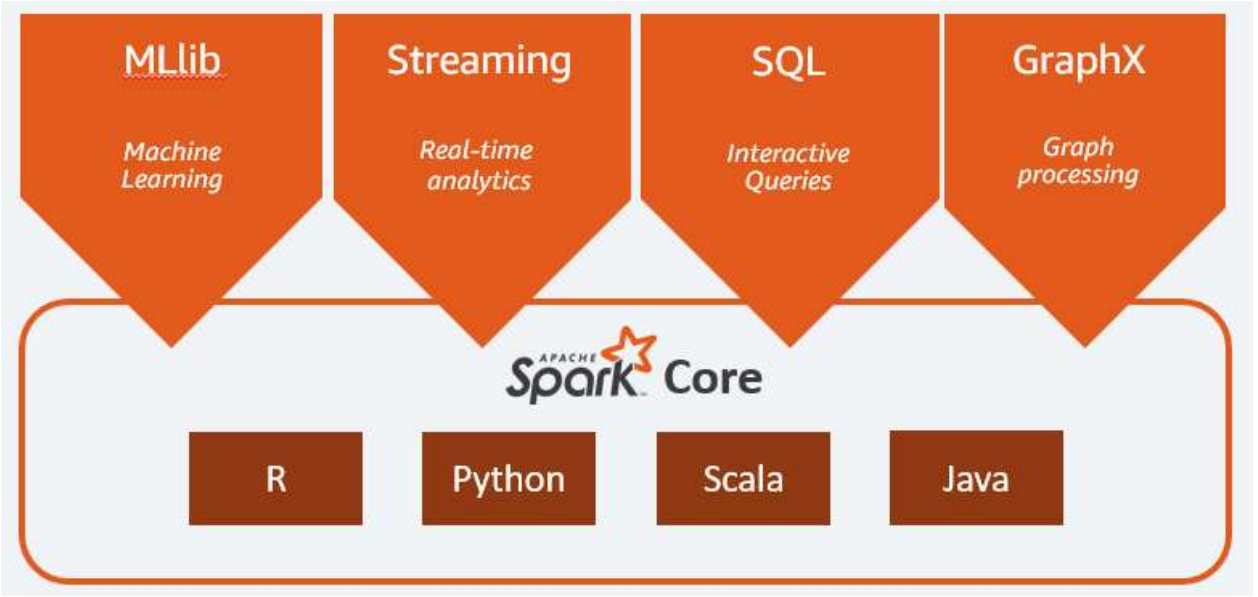
- https://spark.apache.org/
- Unified engine for large-scale data analytics.

Apache Spark (Cont.)

- Is an open-source, distributed processing system used for big data workloads.
- Utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.
- Provides development APIs in Java, Scala, Python and R, and supports code reuse across multiple workloads
  - Batch processing, interactive queries, real-time analytics, machine learning, and graph processing.
- Apache Spark has become one of the most popular big data distributed processing framework with 365,000 meetup members in 2017.

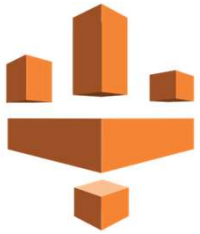
Apache Spark (Cont.)

- Key features
  - Batch/streaming data
  - SQL analytics
  - Data science at scale
  - Machine learning
- Benefits of Apache Spark
  - Fast
  - Developer friendly
  - Multiple workloads



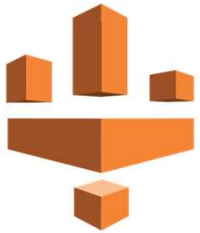
AWS Glue 소개

AWS Glue



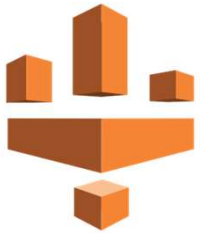
- Is a serverless data integration and ETL service.
- Makes it easy to prepare data for analytics, machine learning, and application development.
- Provides all the capabilities needed for data integration to gain insights and put data to use in minutes instead of months.
- Easily integrating with other AWS data services such as S3, Lambda, and others.
- No infrastructure to set up or manage.
- Pay only for the resources consumed while your jobs are running.
- Is mostly using by Data engineers and ETL developers to create, run and monitor ETL workflows.

AWS Glue (Cont.)



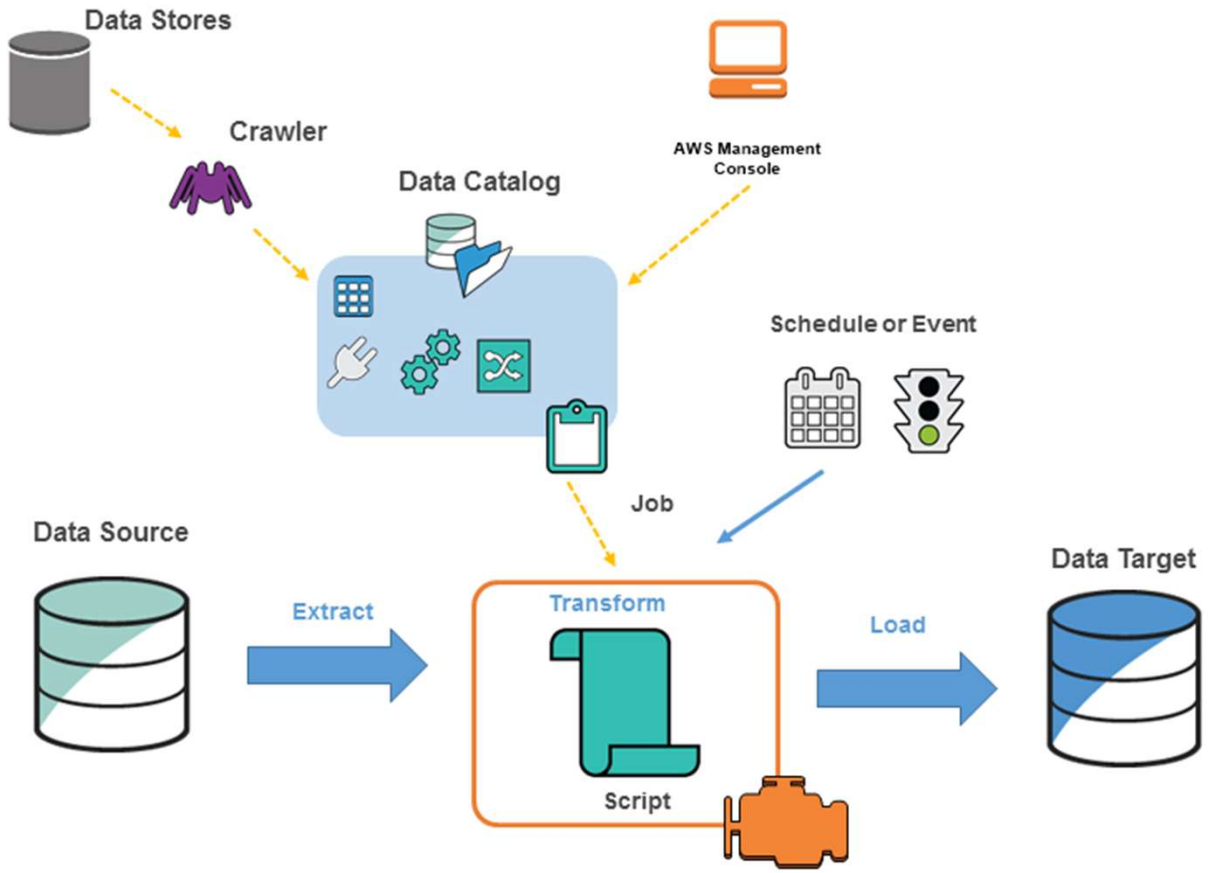
- It has three major components:
  - AWS Glue Data Catalog
  - ETL engine creating Python or Scala code automatically
  - Configurable scheduler
- Refer to
  - https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html
  - https://aws.amazon.com/ko/glue/

AWS Glue (Cont.)



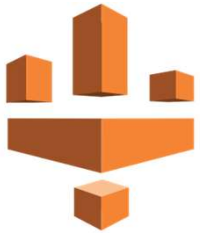
- Why to use
  - **To run serverless queries** across Amazon S3 data lake. ➔ Get right way by getting all the data available at single interface for analysis
  - **To comprehend** your Data Assets. ➔ Data catalog makes job easy to find different AWS data sets. Also saves data in various AWS Services
  - By building event driven ETL workflows, You **can execute ETL operations** once the data is available in Amazon S3 by calling the Glue ETL task from AWS Lambda service.
  - **Useful to organize, clean, verify, and format data** in preparation for storage in a data warehouse or data lake.

AWS Glue (Cont.)



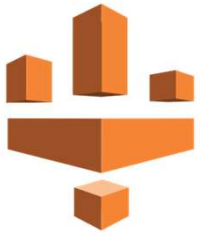
https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html

AWS Glue (Cont.)

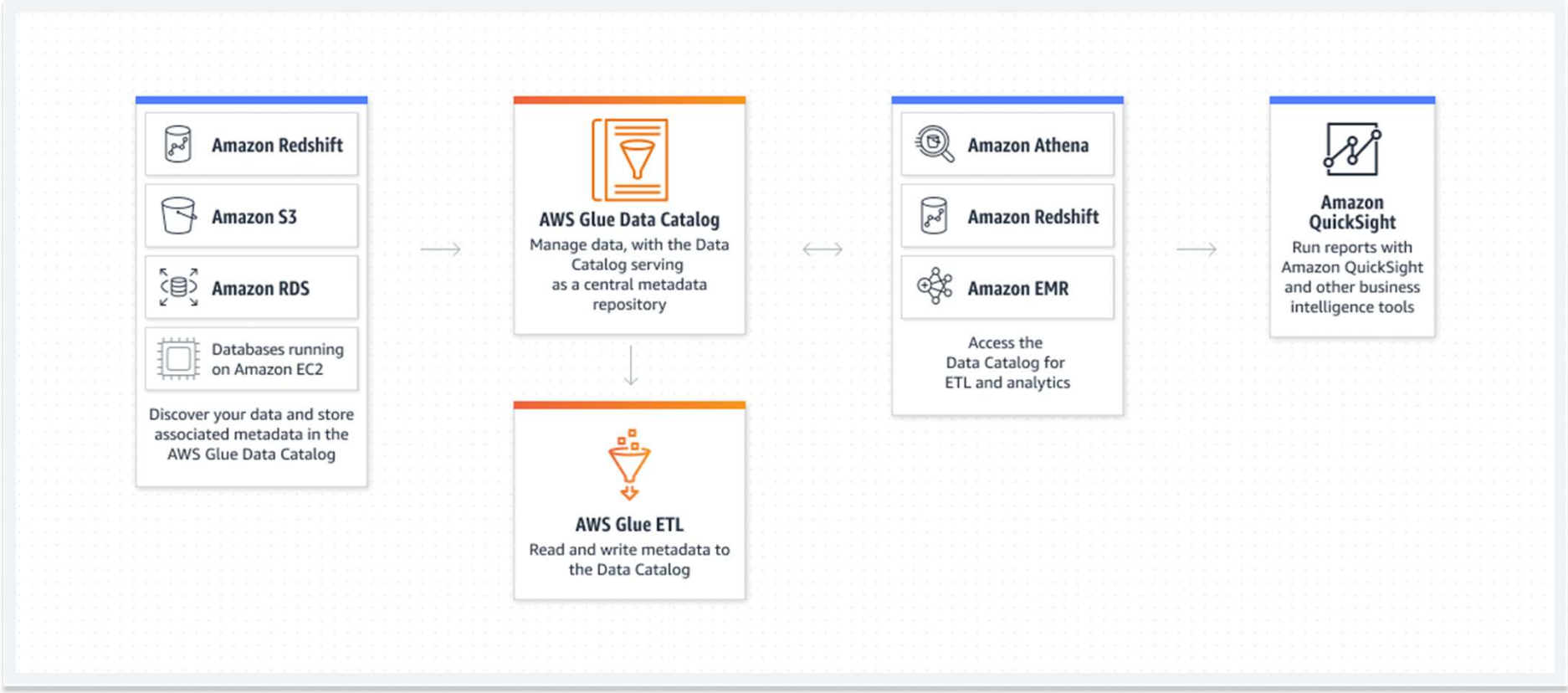


- How it works – Event-driven ETL

AWS Glue (Cont.)

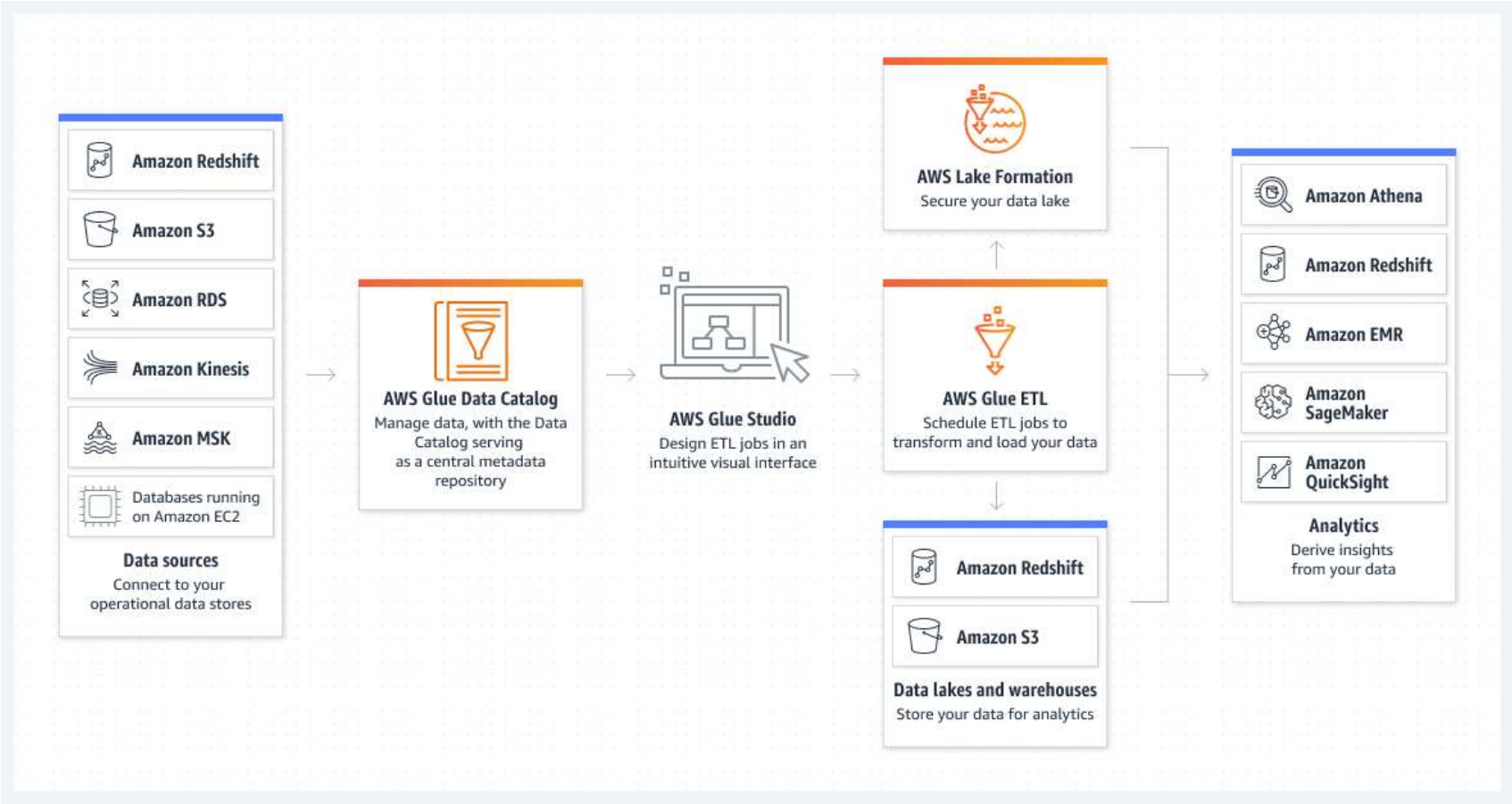


- How it works – AWS Glue Data Catalog



AWS Glue (Cont.)

- How it works – No-code ETL jobs



AWS Glue (Cont.)

- How it works – Data preparation