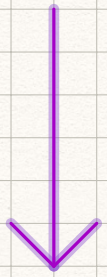
Data Warehouse:

- Mature field, with proved record of performance
- Still useful in many use cases.



Why a Data Lake:

- The abundance of unstructured data (text, xml, json, logs, sensor data, images, voice...)
- Unprecedented data volumes (social, IoT, machine-generated, etc.).
- The rise of Big Data technologies like HDFS, Spark, etc.
- New types of data analysis gaining momentum: graph analytics, predictive analytics...
- Emergence of new roles: Data Scientist.

Abundance of Unstructured Data:

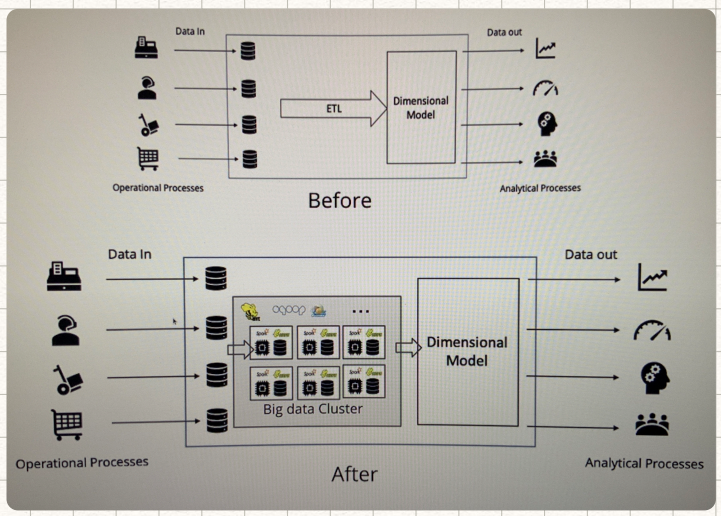
- Might be possible in the ETL process. Distill some elements of file and put them in tabular format
- But Later we might decide we want it to be transformed differently → a particular transformation procedure → Imposes a strong commitment without enough knowledge.
- Some data is hard to transform into tabular format → deep nested JSON
- Some data can be stored as "Blobs" → Remains useless if not processed.

- HDFS reduces cost per TB compared to MPP databases
- Processing Frameworks work on the same machines employed to store data.
- Possibility of performing analysis without requiring previous schema definition.
- Being data a new source of value: Unstructured format applies no "pre-conceptions" to it.

Big Data Technology Effects On Warehouses

Lower Cost & ETL Offloading

- ETL Offloading ; not anymore a required step for data ingestion
- Same hardware for storage and processing
  → no need for a spcial ETL grid.
- Dimensional Modeling for high known value data
- Low cost per TB → Room for storing
  = "low / un know value data" (previously not available)



Schema-on-Read

- Traditionally, data in databases has been much easier to process than data in plain files
- Big Data data tools from Hadoop ecosystem make it easy to work with unstructured data
  - ~~Creating a database~~
  - ~~Inserting data into the database~~
- Schema-on-read → Inferred
  → Specified → when data is read, it is check against the specified schema

Un-Semi Structured Support

- Read and Write: * files in many formats * variety of file system
  * variety of databases (SQL / No-SQL)
- All exposed on a single common abstraction layer.

Data Lakes, Concepts?

Before (No data lake): data was extracted, then transformed in appropriate formats and finally loaded → ETL

Now (Data Lake): data is stored "as-is"; transformations are done later.

Extract - Load - Transform

⊕ Bonus features: - Massive parallelism & scalability come out of the box
- Columnar Storage using Parquet.

Data Lake vs Data Warehouse:

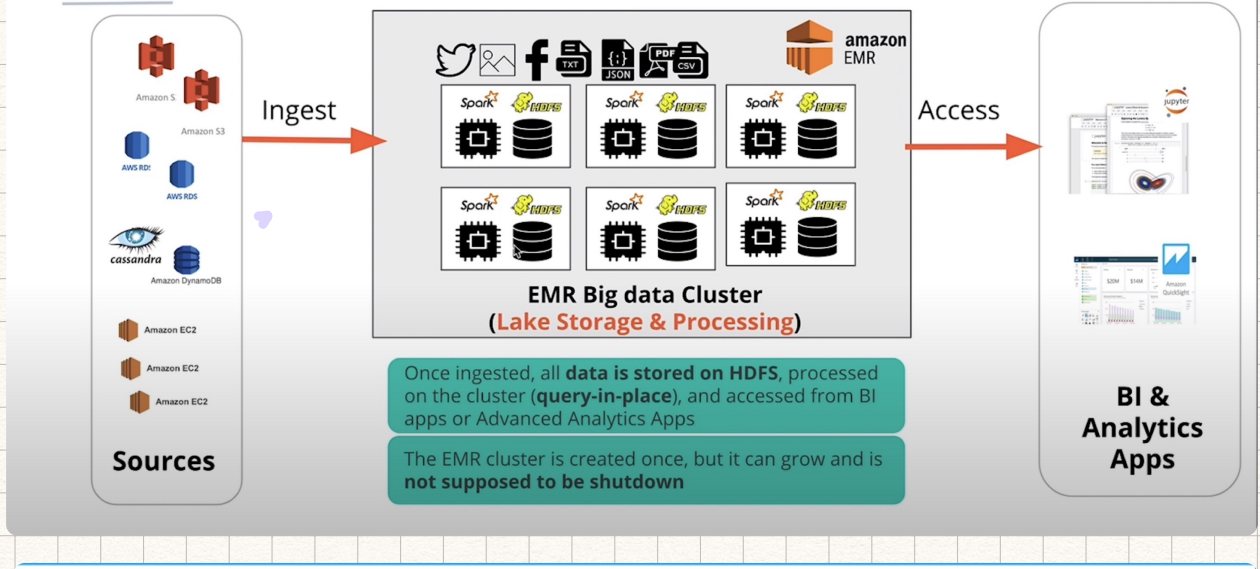
Data Lake vs Data Warehouse

	Data Warehouse	Data Lake
Data form	Tabular format	All formats
Data value	High only	High-value, medium-value and to-be-discovered
Ingestion	ETL	ELT
Data model	Star & snowflake with conformed dimensions or data-marts and OLAP cubes	Star, snowflakes and OLAP are also possible but other ad-hoc representations are possible
Schema	Known before ingestion (schema-on-write)	On-the-fly at the time of analysis (schema-on-read)
Technology	Expensive MPP databases with expensive disks and connectivity	Commodity hardware with parallelism as first principle
Data Quality	High with effort for consistency and clear rules for accessibility	Mixed, some data remain in raw format, some data is transformed to higher quality
Users	Business analysts	Data scientists, Business analysts & ML engineers
Analytics	Reports and Business Intelligence visualizations	Machine Learning, graph analytics and data exploration

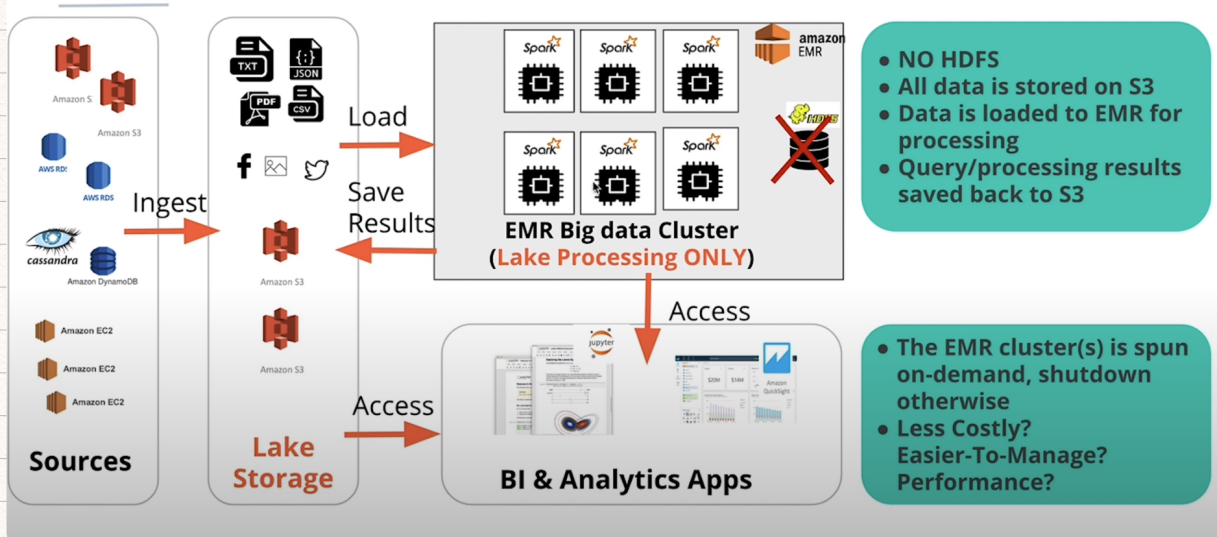
Data Lake options on AWS:

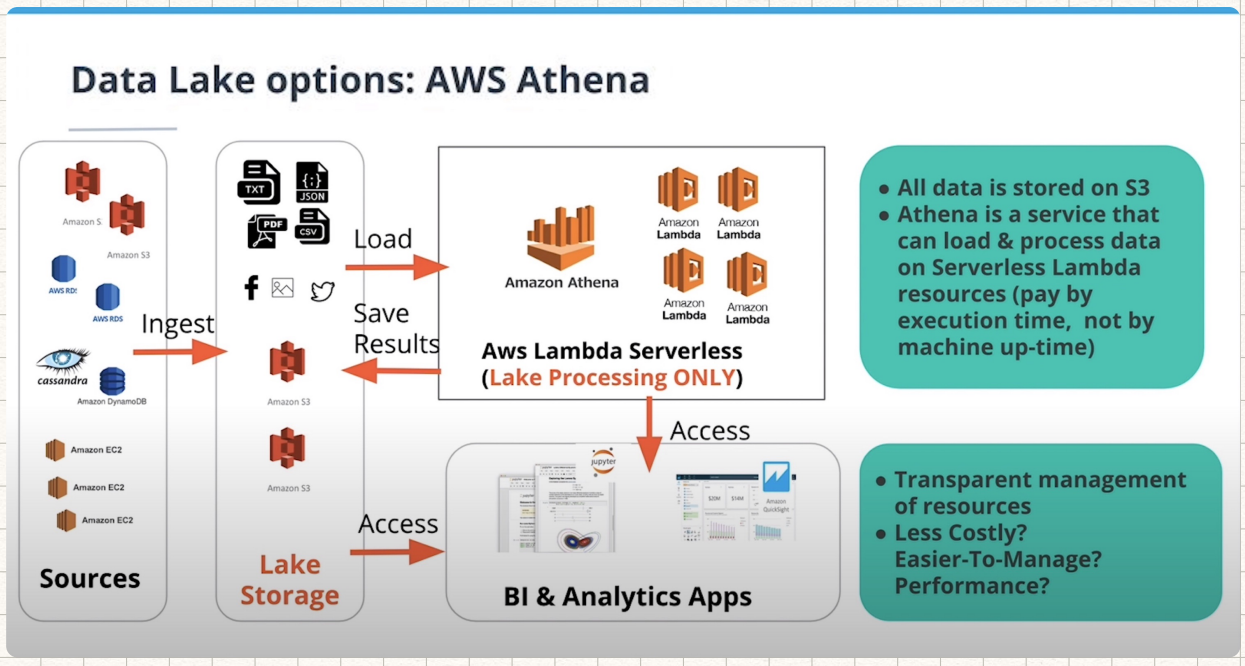
HDFS	Spark	AWS EMR (HDFS+Spark)	EC2 + Vendor Solution
S3	Spark	AWS EMR (Spark)	EC2 + Vendor Solution
S3	Serverless	AWS Athena	Lambda + Vendor Solution
"Storage"	"Processing"	"AWS-Managed Solution"	"Vendor Managed"

Data Lake options: AWS EMR (HDFS+Spark)



Data Lake options: AWS EMR (S3+Spark)





Data Lake Issues

- Data Lake can prove to be a chaotic garbage dump
  → Practices like detailed metadata to reduce risk.
- Since data lakes allow wide accessibility of cross-department data and external data.
  → Sometimes data governance is not easy to implement.
- Sometimes hard to know if a data lake should replace a data warehouse.