SketchNotes By Nived Varma

Traditional Data Warehouse

- Uses structured, relational tables.
- Data is queried using SQL
- Eforcement of rigid schema-on-write (Data must confirm to predefined structure)
- Optimized for structured data & complex queries
- Generally more expensive for large-scale storage.

Data Lake Evolution

- Emerged as a response to "big data" challenge
- 3 V's
  - Volume:
  - Variety
  - Velocity
- Stores data in raw format.
- Schema-on-Read (No fixed schema)
- More cost effective for large scale storage
- Flexible but rather hard to query.

Lakehouse

Combines Benefits of both DW & DL.

Data Lakehouse (modern Approach)

files are stored in Table as Parquet Formats

Microsoft Fabric

Storage Layer: OneLake (ADLS Gen2)

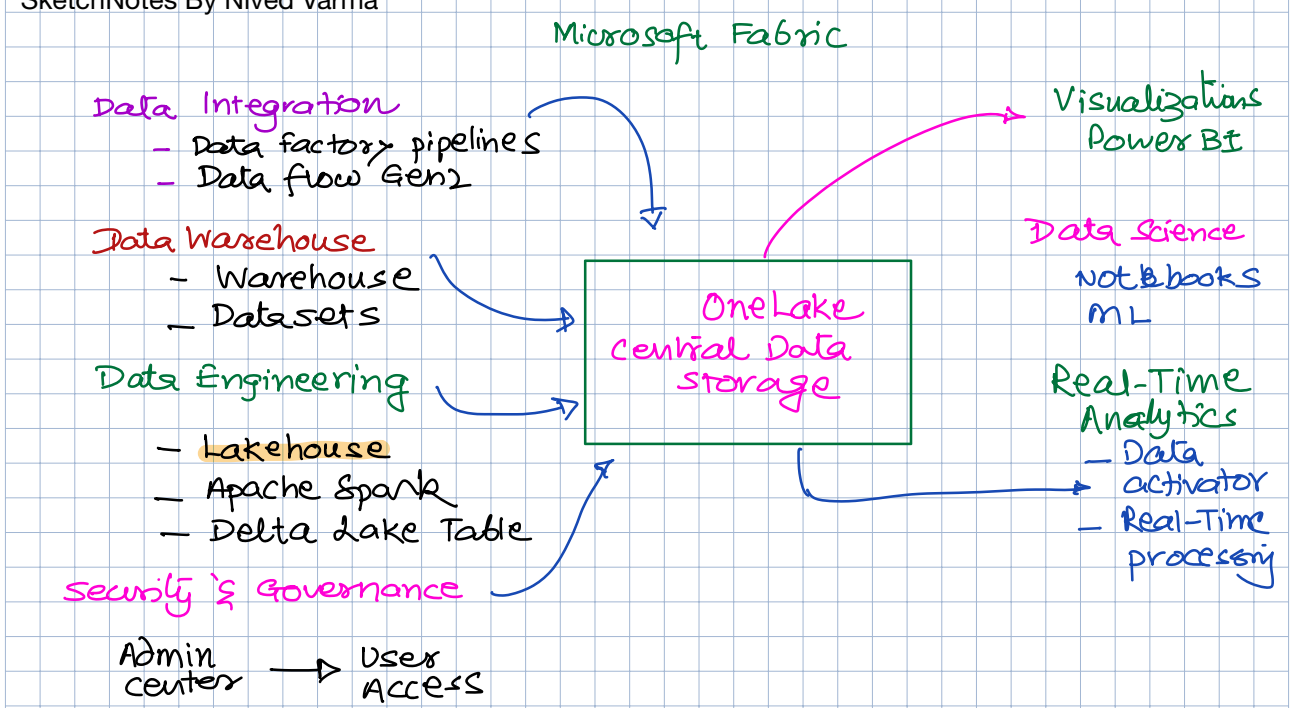
Metadata Layer: Uses Delta Lake format

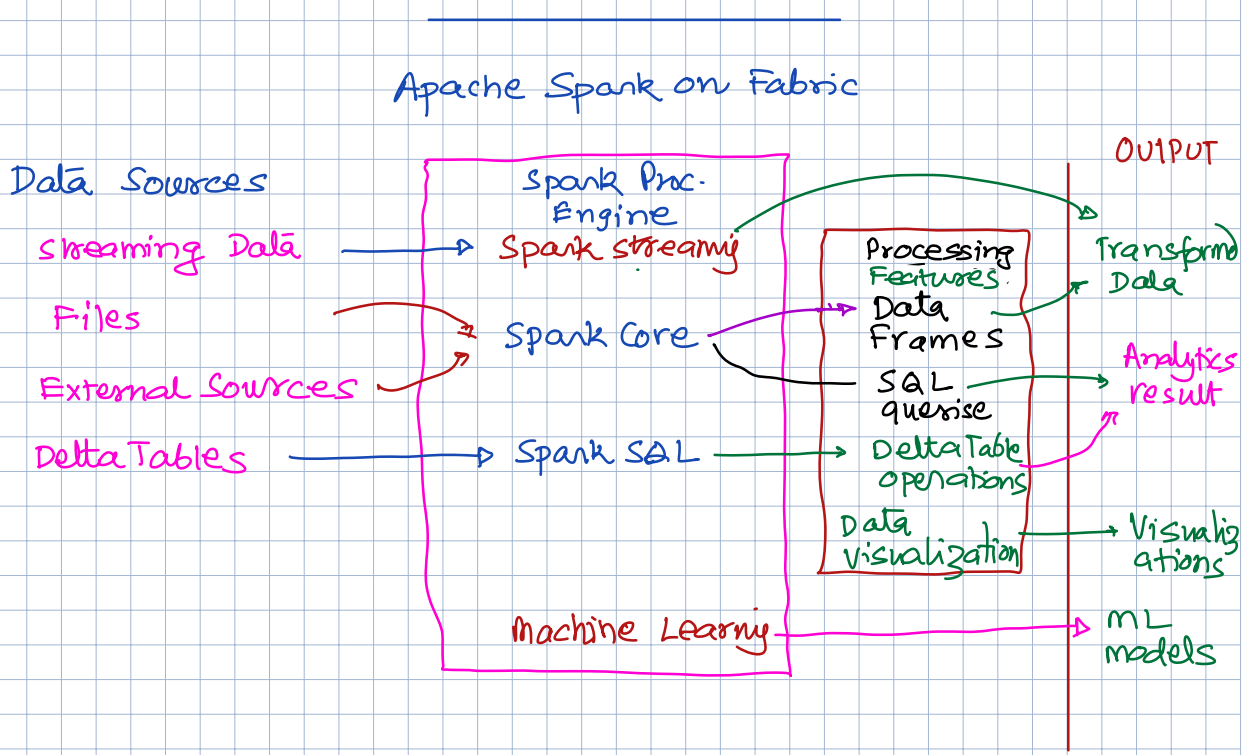
- schema enforcement
- ACID transaction
- Data Versioning (Time Travel)
- schema evolution
- SQL Query capabilities

Benefits

- Allows raw file storage & structured querying
- More flexibility
- More structured than pure data Lake.

SketchNotes By Nived Varma





SketchNotes By Nived Varma

Spark configuration

step-1 Create a Fabric WS → step-2 Enable spark compute → step-3 configure Spark settings

Key Config. Settings

Compute setting

- ✓ Executor size
- ✓ No. of executors
- ✓ Drive Size
- ✓ Spark Version
- ✓ Runtime version

Small (4 cores) med (8 cores) Large (16 cores)

Parallel Processing Units

Controls coordination of tasks

Memory Settings

- ✓ Executor memory
- ✓ Driver Memory
- ✓ Memory overhead
- ✓ Shuffle Memory
- ✓ Cache Memory

4–8 GB/executor

Important for collecting results

Advance settings

- ✓ Dynamic allocation
- ✓ Custom packages
- ✓ Env. Variables
- ✓ Spark properties
- ✓ Logging

Automatic allocation of executors

Third party dependencies

SketchNotes By Nived Varma

Spark Notebooks Vs Jobs.

Exploratory work

Notebook

- ✓ Data Exploration
- ✓ Model Development
- ✓ Documention

Production-ready scheduled execution.

Jobs

Production ETL
- ✓ scheduled data processing
- ✓ Large-scale transformations
- ~ Automated workflows.

Batch Processing

Model Deployment

Key Benefits Using Spark

- **Integration** Works with many tools
- **Performance** upto 100x faster than traditional
- **Flexibility** Multiple Data formats & sources
- **Scalability** Handle large data volumes

SketchNotes By Nived Varma

Spark Data Frame

ID	Name	Age	City
1	Vijay	34	Blore
2	Neha	28	Chennai
3	Amit	36	Hyderabad

- ✓ Distributed collection of data organized into names & columns
- ✓ Immutable & lazily evaluated
- ✓ Supports – structured as well semi-structured data.

Original Data Frame
ID
Name
Age
City

Filter
df.filter(age>25)
df.where(city == 'chennai'

Select
df.select('name','age')
df.selectExpr('age+1')

Group By
df.groupBy('city').
.agg(avg('age')

JOIN
df1.join(df2, df1.id == df2.id)

Output → DF