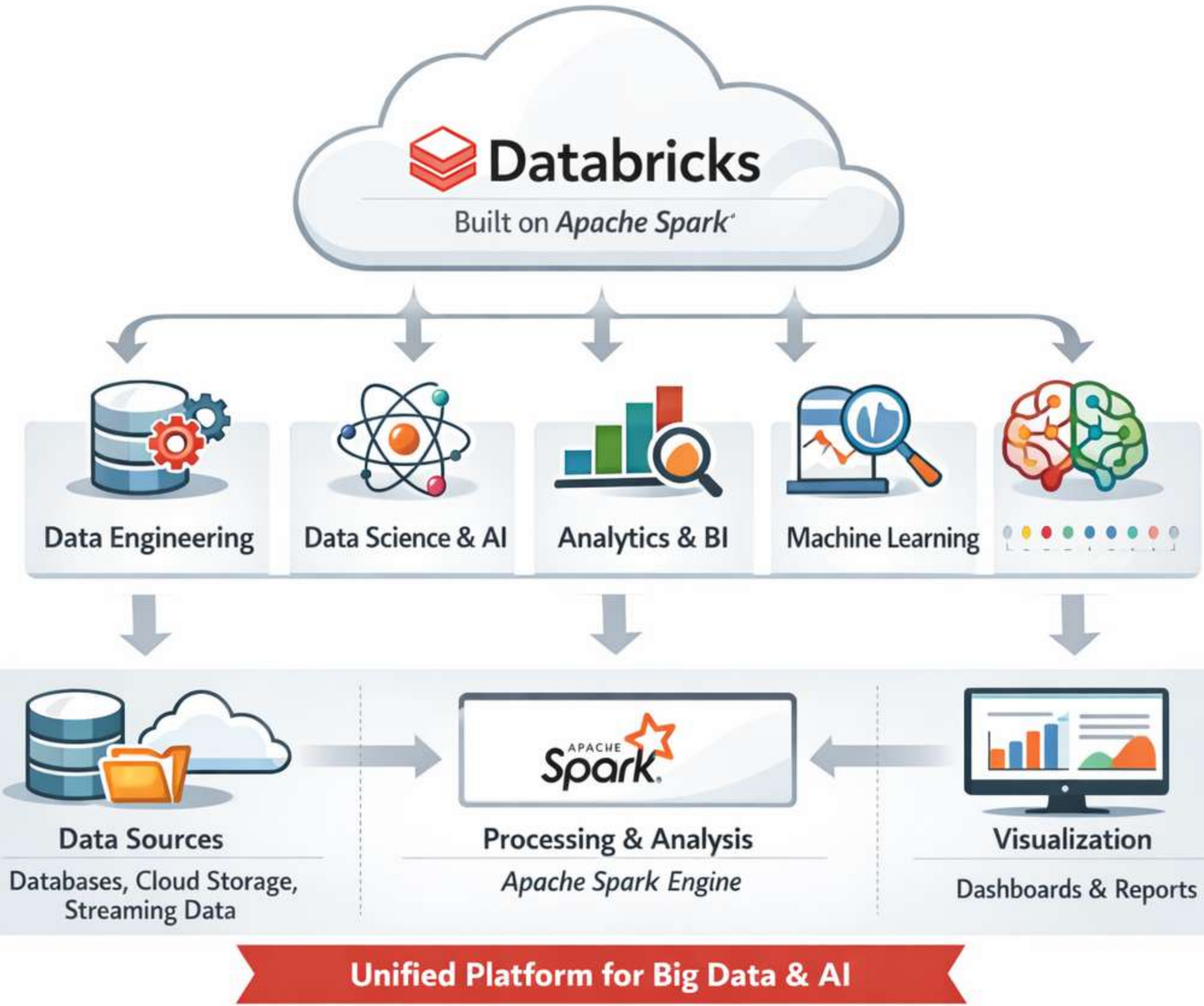
databricks

Unified Data Platform for Big Data & AI



Databricks is Needed?

Handles Big Data Efficiently

Apache Spark processes large-scale data quickly

All-in-One Platform

Combines Data Engineering, Analytics, Machine Learning

Databricks

Built on *Apache Spark*

Faster Analytics & Insights

In-memory processing makes everything faster

Easy Collaboration

Using Notebooks & Shared Data

Scales Automatically

Cloud-based & scales up/down automatically

Scales Automatically

Cloud-based & scales up/down automatically

Supports Advanced Analytics & AI

For Real-time Analytics & Machine Learning

Unified Data Platform for Big Data & AI

How Databricks Handles Scale

Data Sources
(Files | Databases | Streams)

Databricks
(Apache Spark Engine)

Auto-scaing Clusters
Add / Remove cdmpe as needed

Auto-Scaling Clusters
Scale compute up/down as needed

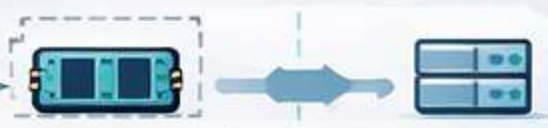
Auto-Scaling Clusters
(Add / Remove Compute Nodes)

High workload
More nodes (high power)

Databricks
(Apache Spark Engine)

Cloud-Based Compute
aws | Azure | GCP

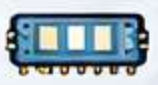
Distributed Processing
- Split data into smaller chunks
- Process chunks in parallel across nodes
- More machines = faster processing



Data cached in RAM (very fast)

Cloud-Based Compute
aws | Azure | GCP

In-Memory Processing
Data cached in RAM (very fast)



Distributed Processing
- Split data into smaller chunks
- Process chunks in parallel across nodes
- More machines = faster processing

Storage ≠ Compute
Separation of compute dusters and storage (Data Lake)

Auto Scaling Clusters
Scale compute up/down

Cloud-Based Compute
Runs on AWS, Azure, GCP

Clusters & DataFrames in Databricks

Distributed Processing with Apache Spark

Databases

Cloud Storage
(S3, ADLS, GCS)

Streaming Data

Auto-Scaling Cluster

Databricks

Built on *Apache Spark*

Master

Auto-Scaling Cluster

Apache Spark Clusters
(S3, ADLS, GCS)

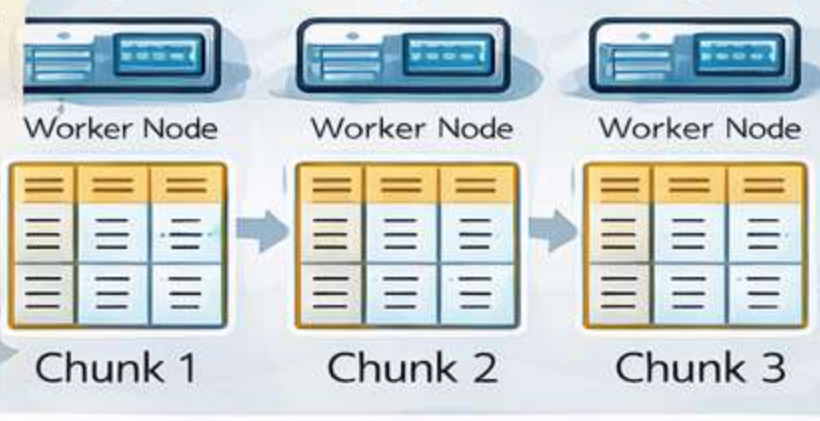
Worker Node

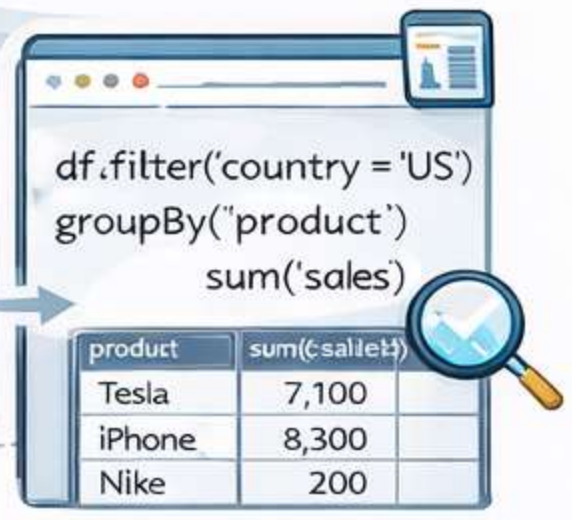
Worker Node
Master Node

Auto-Scaling Cluster

Fast Results
(Analytics / ML)

DataFrame:
Distributed table stored in memory, split into smaller "chunks"





Distributed Processing

Breaks big data into "chunks" and processes them in parallel across a cluster

DataFrames in RAM

Keeps chunks in-memory for high-speed queries

Auto-Scaling Clusters

Adds or removes worker nodes based on workload

Lazy Execution in Databricks

(Apache Spark)

Spark builds an **execution plan first and runs** it only when a result is required.

Data Source

- df.select()
- df.filter()
- df.groupBy()

(Transformations)

Databricks

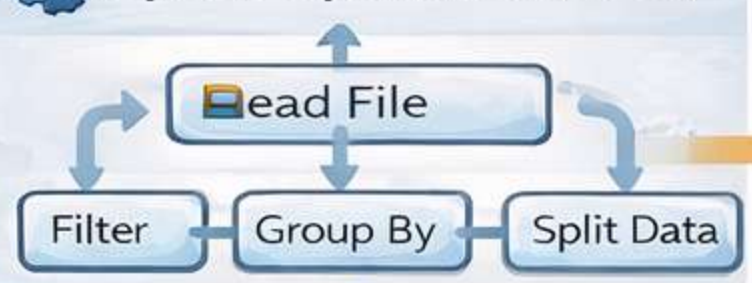
Built on *Apache Spark*

Action Triggered

df.show()
df.count()
df.collect()

Runs eveything now

Spark Optimizer (DAG)



Runs everything now

Optimized Execution

Result Output

Transformations (Lazy)

- select()
- filter()
- groupBy()
- withColumn()

Sparks records steps

Actions (Trigger Execution)

- show()
- write()
- count()

Spark runs all steps now

Databricks SQL & Tables

Use SQL Queries

```
SELECT product, SUM(sales)
FROM sales
JOIN products ON
sales.product_id = products.id
WHERE country = 'US'
GROUP BY product
ORDER BY SUM(sales) DESC;
```

Analyze & Transform Data in Tables

product	sales	Growth
Tesla	5,200	20%
iPhone	6,800	15%
Nike	320	5%

UPDATE INSERT DELETE

Build Fast Analytics on a Data Lake

Databricks combines fast analytics with data lake storage, enabling powerful **data analysis and transformation** with SQL.

AI & Why This Matters?

1 Better Decisions

AI analyzes data to find patterns and insights

2 Faster Results

Queries and predictions much quicker

Databricks

Unified Data & AI Platform

4 Cost Savings

Automating tasks reduces costs & errors

3 Innovation & Growth

AI drives new products and better services

Databricks

4 Cost Savings

Automating tasks reduces costs & errors

Innovation & Growth

AI drives new products and better services

Databricks combines AI accessible by **simplifying large-scale analytics, machine learning, and data engineering with SQL.**

Why Databricks?

Easily train and **deploy AI models** on your data