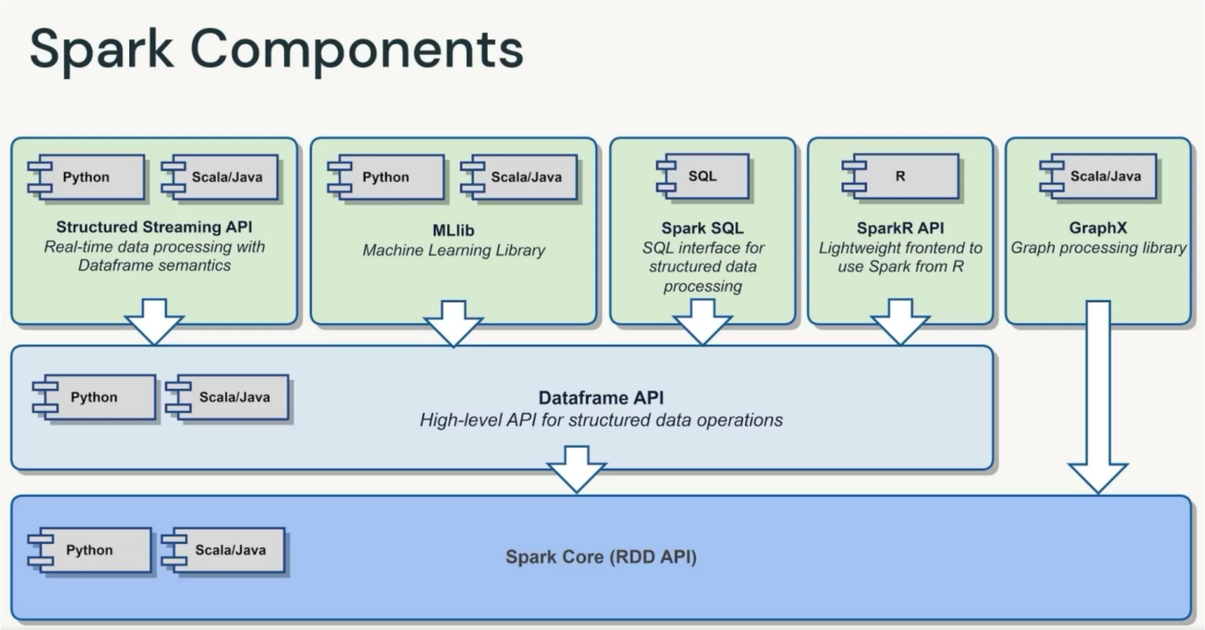
Apache Spark

Apache Spark is an open source, distributed, unified analytics engine for fast, last-scale data processing.

- Supports SQL, Structured Streaming, Machine Learning, and Graph Processing.
- Performs In memory computing.

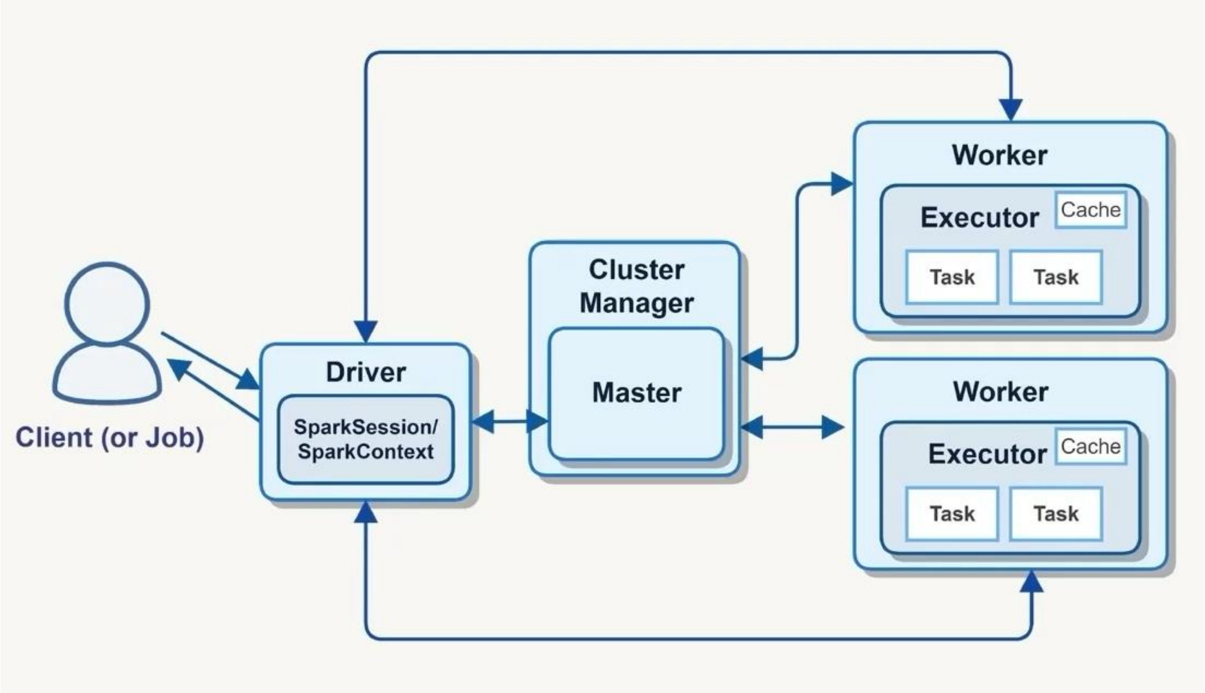
Components



The core engine provides foundation for all applications, this includes memory management, fault tolerance & recovery, and scheduling tasks distribution.

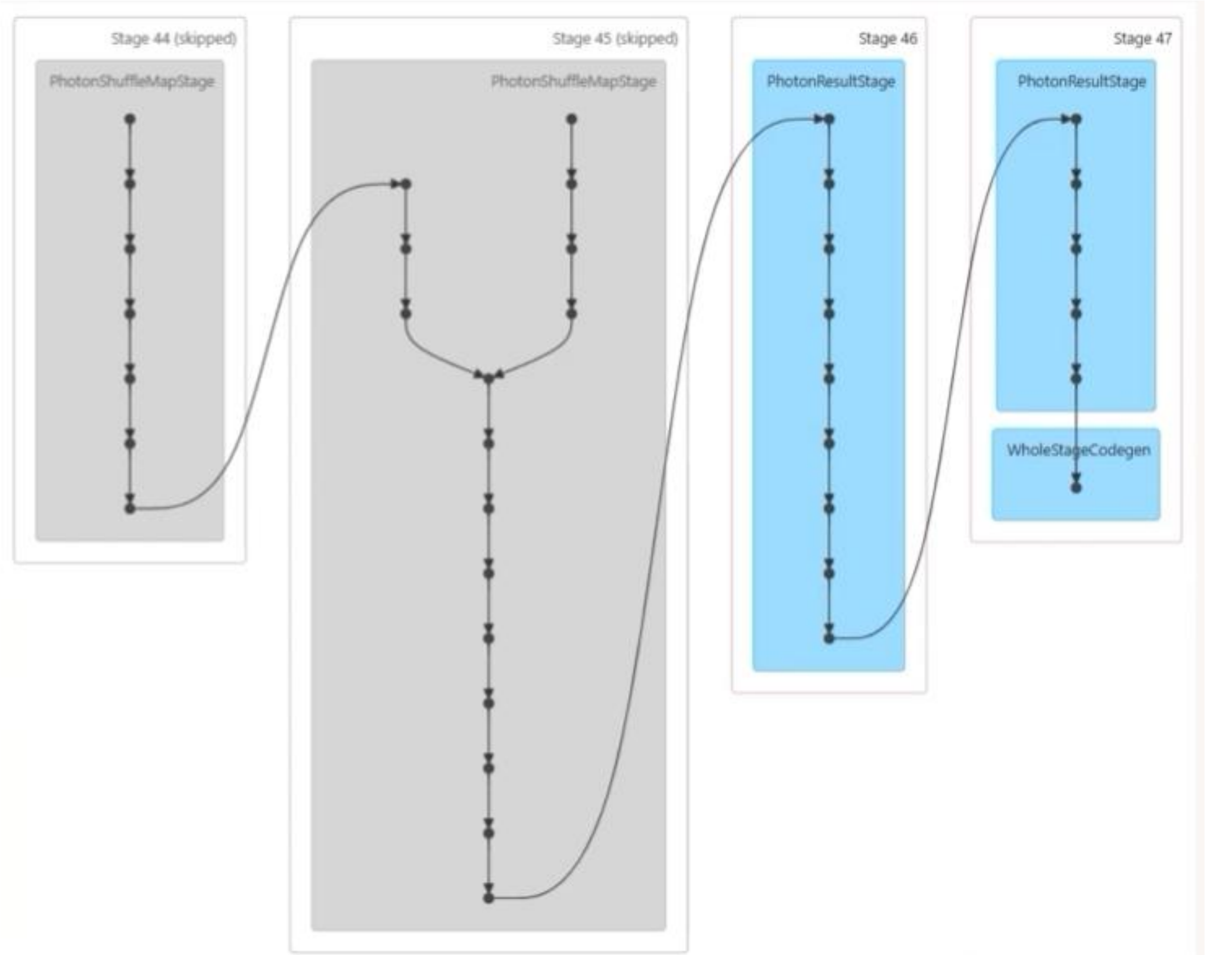
This core engine is exposed to higher level APIs like Dataframe, Structured Streaming, MLib, and GraphX which are then exposed to different languages like scala and python.

Architecture



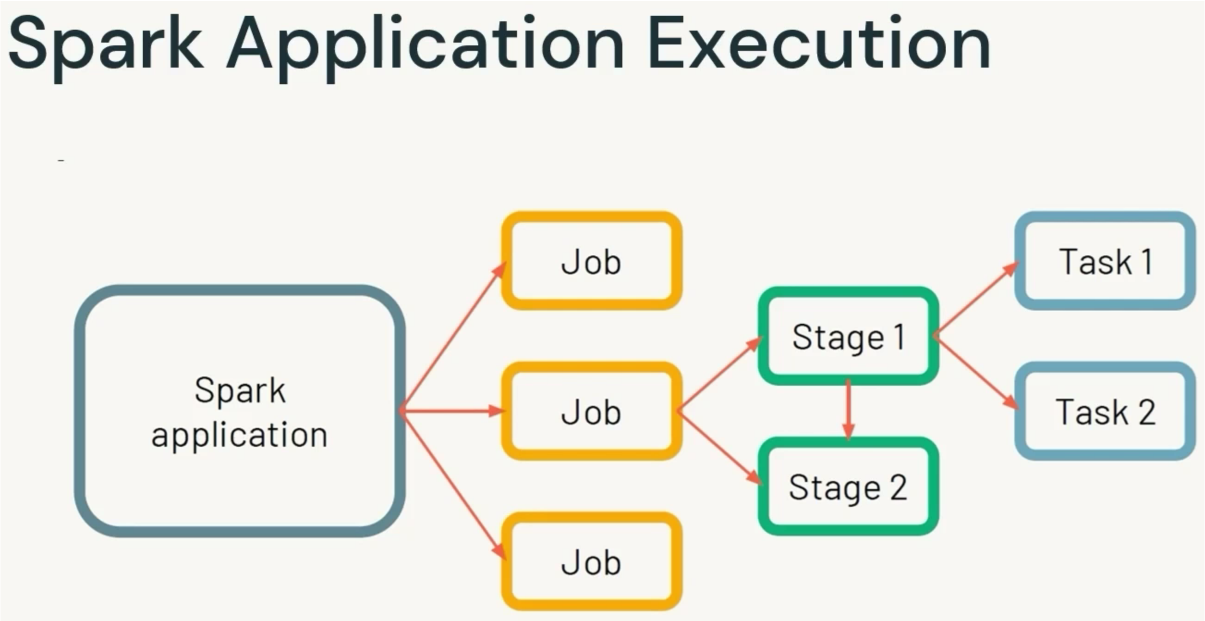
Sparks follows Client-Server Architecture and have the following components

1. **Driver:** Driver is the centralized entry point for the spark application it is responsible for managing and optimizing the execution. (this can be a query or a job).
    a. This creates Spark Session.
    b. Analyses the spark application and constructs a DAG.
    c. Schedules and distributes tasks to executors for execution.
    d. It also monitors these executions and handles failures.
    e. Finally returns the results to the client.
2. **Cluster Manager:** Manages resources and allocates them to the Driver.
3. **Workers:** are the Nodes (VMs) that host executors. A worker can have multiple executors.
4. **Executors:** run tasks and cache data. So, these are ones which do heavy lifting.
    a. These can be configured based on requirements like number of CPU cores and memory.
    b. These stores intermediate and final results in memory or disk.



Spark application is implemented using DAGs meaning tasks flow in one direction.

Spark Jobs are broken down into stages. Stages are the group of tasks that can be executed in parallel.



Spark Application Execution

1. An application can trigger multiple jobs.
2. Jobs are high level operations which will be trigged by actions.

3. Jobs are implemented by DAGs.
4. These jobs are divided into stages.
5. Stages are groups of tasks that can be executed in parallel.
6. Stages communicate with each other with a process called shuffle process.

The tasks within a stage run in parallel, Know as **Shared Nothing** mode.

Databricks Compute types

1. **All purpose:** For developers running experiments in notebooks
2. **Job clusters:** For ephemeral tasks like ETL.
3. **SQL warehouse:** For BI End points.

Dataframe API