ASSESSMENT - 17 | PySpark - 02

*“APACHE SPARK, PYSPARK ARCHITECTURE & CLUSTER MANAGER”*



Submitted By

VAIBHAV PATIDAR

IPS ACADEMY, INDORE (M.P.)

**Date :** 22-12-2023 | **Day 17**

**Week 3** | **Day - 05**

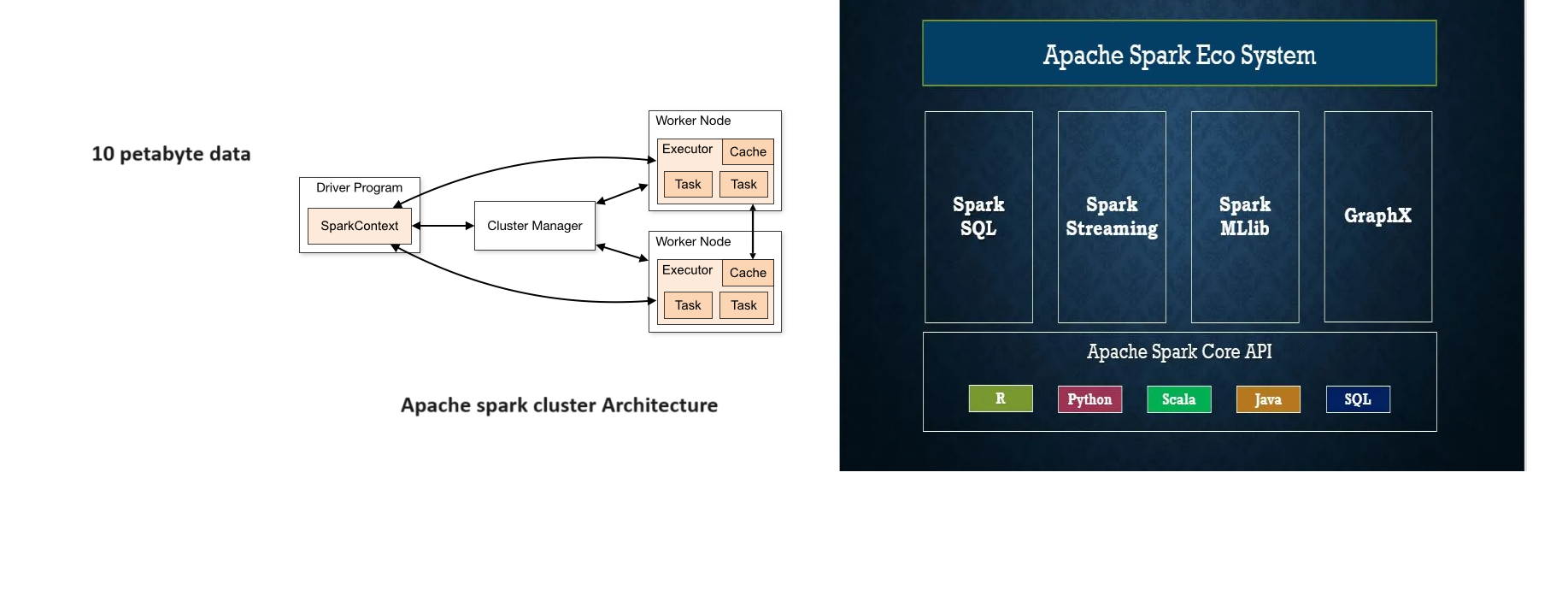
**→ What is Apache Spark?**

* Apache Spark is an open-source, distributed computing system that is designed for big data processing and analytics.
* Spark is known for its speed. It can process data in memory, making it faster than traditional data processing tools like Hadoop.
* Spark distributes data processing tasks across a cluster of computers, enabling parallel processing and efficient use of resources.
* Spark supports multiple programming languages, including Java, Scala, Python, and R
* RDDs are the fundamental data structure in Spark. They are distributed collections of data that can be processed in parallel.
* Spark includes a module for structured data processing called Spark SQL. It allows users to execute SQL-like queries on structured data using Spark.
* Spark provides a machine learning library called MLlib, which offers a set of tools and algorithms for machine learning tasks.
* Spark includes a graph processing library called GraphX, which facilitates the analysis of graph-structured data.
* Spark supports real-time data processing through Spark Streaming, allowing the processing of live data streams.

**→ What is PySpark?**

* **Python Library for Spark:** PySpark is a Python API for Apache Spark, facilitating interaction with Spark's distributed computing capabilities.
* **Big Data Processing:** Enables Python developers to process large-scale datasets using Spark's parallel and distributed computing model.
* **Spark Context:** Provides a Python-friendly interface through a SparkContext object, connecting to Spark clusters for communication.
* **DataFrames:** Introduces DataFrames, distributed data collections with named columns, for efficient manipulation and analysis.
* **Integration:** Seamlessly integrates with Python libraries, supporting data analysis, machine learning, and visualization.

**→ PySpark Architecture :**

****

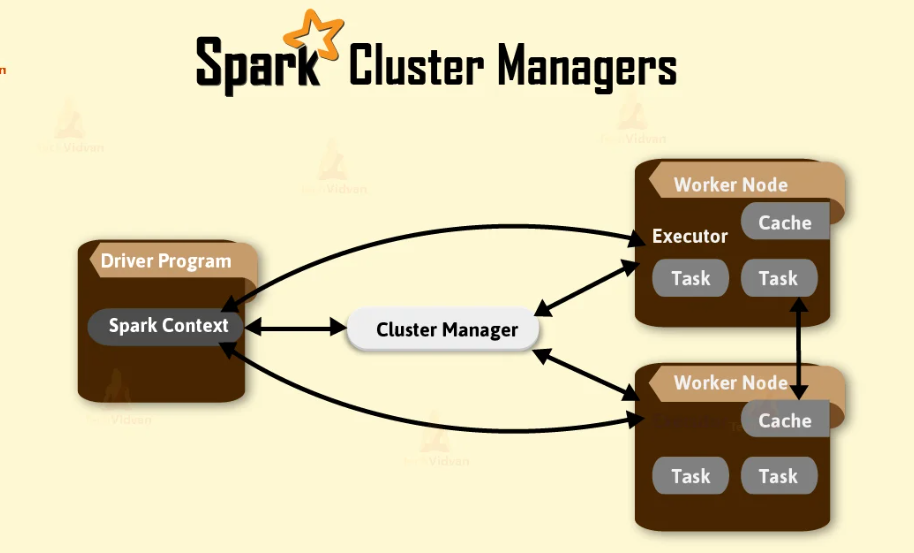
* **Driver Program:** Initiates PySpark applications, coordinating tasks and managing resources.
* **Spark Context:** Communicates with the cluster manager, distributing tasks to worker nodes.
* **Cluster Manager:** Coordinates resources and manages the allocation of tasks across the cluster.
* **Worker Nodes:** Execute tasks in parallel, processing data and returning results to the driver program.
* **RDDs (Resilient Distributed Datasets):** Fundamental data structures distributed across the cluster, enabling fault-tolerance and parallel processing.
* **Transformations and Actions:** Transformations modify RDDs, while actions trigger computations and return results.
* **DataFrames:** Abstraction over RDDs, providing structured data processing capabilities for ease of use.

**→ PySpark Features :**

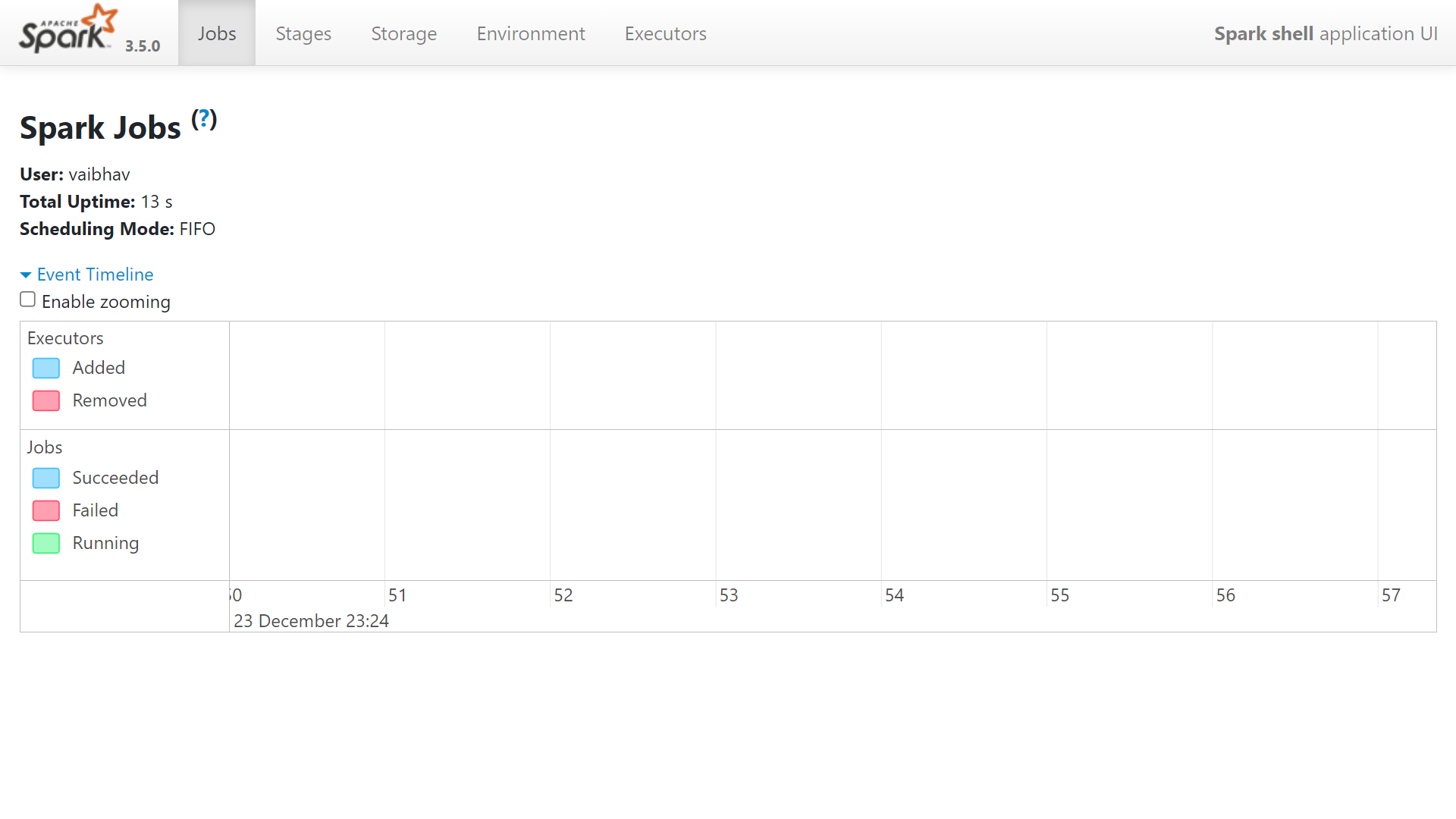
* In-memory computation
* Distributed processing using parallelize
* Can be used with many cluster managers (Spark, Yarn, Mesos e.t.c)
* Fault-tolerant
* Immutable
* Lazy evaluation
* Cache & persistence
* Inbuild-optimization when using DataFrames
* Supports ANSI SQL

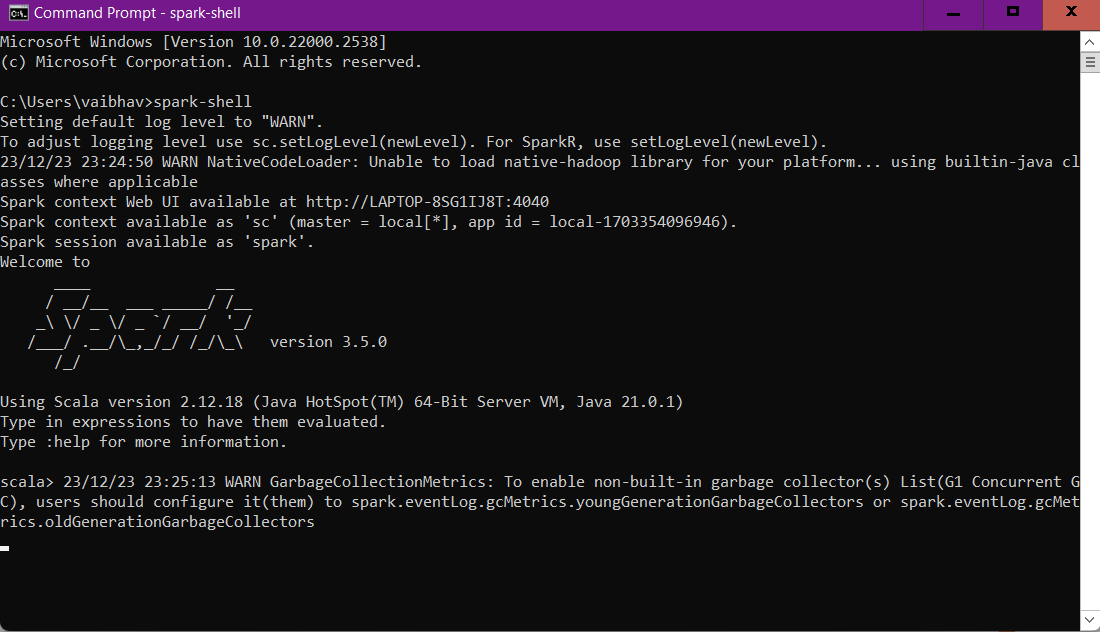
**→ Applications of Cluster Manager in PySpark :**

* **Cluster Management:** Cloud Manager in PySpark is a tool for managing Spark clusters on cloud platforms like AWS, Azure, or Google Cloud.
* **Resource Allocation:** It helps allocate resources efficiently, ensuring optimal performance and scalability for Spark applications.
* **Scaling:** Cloud Manager allows for easy scaling of clusters based on workload demands, adding or removing resources as needed.
* **Integration:** Seamlessly integrates PySpark with cloud services, streamlining the deployment and management of Spark applications in cloud environments.



**→ Spark Web UI :**

****

****