Single node/Multiple node Hadoop:

Hadoop can be set up in either a single-node or multiple-node configuration. In a single-node setup, Hadoop runs on a single machine, where all Hadoop components, such as the Hadoop Distributed File System (HDFS) and MapReduce framework, are installed and executed on the same machine. This configuration is suitable for development, testing, or learning purposes.

On the other hand, a multiple-node Hadoop setup involves distributing the Hadoop components across multiple machines in a cluster. This distributed configuration allows for increased processing power, storage capacity, and fault tolerance. It enables parallel processing of large datasets by dividing the workload among the nodes in the cluster.

What is jps:

"jps" stands for **Java Virtual Machine (JVM) Process Status.** It is a command-line utility that comes with the Java Development Kit (JDK) and is used to display the list of Java processes running on a machine. In the context of Hadoop, running the "jps" command shows the status of various Hadoop daemons, including the NameNode, DataNode, ResourceManager, NodeManager, and others, allowing you to verify if the Hadoop services are running properly.

How to start Hadoop:

To start Hadoop, you need to follow these general steps:

1. Ensure that Hadoop is properly installed and configured on your machine or cluster.
2. Start the Hadoop daemons by running the appropriate startup scripts. For example, you can use the "start-all.sh" script to start all the Hadoop daemons simultaneously.
3. Verify the status of Hadoop daemons using the "jps" command. It should show the running daemons.
4. Once the Hadoop daemons are running, you can interact with Hadoop using various command-line tools or by submitting MapReduce jobs.

What is Hadoop:

Hadoop is an open-source framework that provides a distributed computing platform for processing and storing large datasets across a cluster of computers. It consists of two main components:

1. Hadoop Distributed File System (HDFS): HDFS is a distributed file system that stores data across multiple machines in a cluster. It provides high-throughput access to data and ensures fault tolerance by replicating data across different nodes.
2. MapReduce: MapReduce is a programming model and processing paradigm used to process and analyze large datasets in parallel across a Hadoop cluster. It divides the input data into smaller chunks and distributes them to different nodes in the cluster for processing.

Explain MapReduce:

MapReduce is a programming model and computational paradigm designed for processing and analyzing large datasets in parallel across a distributed computing cluster. It consists of two main phases:

1. Map Phase: In this phase, the input data is divided into smaller chunks and processed independently by multiple map tasks running on different nodes in the cluster. Each map task performs a transformation operation on its portion of the input data and generates intermediate key-value pairs.
2. Reduce Phase: In this phase, the intermediate key-value pairs generated by the map tasks are shuffled, sorted, and grouped based on their keys. The reduce tasks then process the grouped data and perform aggregations or computations to generate the final output.

The key idea behind MapReduce is the parallel processing of data by dividing it into smaller chunks and distributing them across multiple nodes. This approach enables scalable and efficient processing of large datasets.

Use of MapReduce:

MapReduce is widely used for large-scale data processing tasks, such as:

1. Batch Processing: MapReduce allows you to process massive datasets in parallel, making it suitable for various batch processing tasks, including log analysis, data cleansing, ETL (Extract, Transform, Load), and generating reports.
2. Data Transformation and Aggregation: MapReduce enables you to perform complex transformations

Data cleaning: Data cleaning, also known as data cleansing or data scrubbing, refers to the process of identifying and correcting or removing errors, inconsistencies, and inaccuracies in a dataset. It involves handling missing values, outliers, duplicate entries, incorrect formats, and other anomalies in the data.

Use of data cleaning: Data cleaning is crucial in ensuring the accuracy, reliability, and quality of data used for analysis, decision-making, and reporting. By cleaning the data, you can eliminate errors that could lead to incorrect insights or misleading conclusions. It helps to improve the integrity of the dataset and enhances the reliability of any downstream analysis or applications.

Functions used for data cleaning: Various functions and techniques are used for data cleaning, such as:

1. Handling missing values: Imputing missing values using techniques like mean, median, or interpolation.
2. Removing duplicates: Identifying and removing duplicate entries from the dataset.
3. Standardizing formats: Ensuring consistent formats for dates, addresses, names, etc.
4. Outlier detection: Identifying and handling outliers that may affect analysis results.
5. Correcting data types: Ensuring appropriate data types for variables (e.g., numeric, categorical).
6. Data validation: Verifying data against defined rules or constraints to identify discrepancies.

Data integration: Data integration refers to the process of combining data from multiple sources or systems into a unified and coherent view. It involves consolidating data from different formats, structures, or databases to create a comprehensive dataset that can be used for analysis or reporting purposes.

Difference between data integration and merging of data: Data integration involves combining and harmonizing data from different sources to create a unified view. It focuses on resolving schema conflicts, handling inconsistencies, and ensuring data interoperability. On the other hand, merging of data typically refers to the simple combination of datasets based on common fields or keys, without necessarily addressing data quality or schema compatibility.

Use of data transformation: Data transformation involves converting or reshaping data from one format or structure to another. It may include operations such as aggregating, summarizing, filtering, or reformatting data to meet specific requirements for analysis, modelling, or visualization.

Error correction: Error correction refers to the process of identifying and rectifying errors or mistakes in the dataset. It involves detecting and resolving inaccuracies, inconsistencies, or anomalies that may have occurred during data collection, storage, or processing.

Data modelling: Data modelling is the process of creating a conceptual representation of data to understand its structure, relationships, and business rules. It involves designing a schema or blueprint that defines how data elements are organized, stored, and accessed. Data modelling helps in capturing the essence of the real-world domain and provides a foundation for database design, data integration, and application development.

**what is subset of Dataset ? in simple terms**

A subset of a dataset refers to a smaller portion or selection of the original dataset. It is created by choosing specific rows or columns from the original dataset based on certain criteria or conditions. The subset contains a subset of the original data, usually based on a specific subset criterion or filtering condition.

For example, imagine you have a large dataset containing information about customers, including their names, ages, locations, and purchase histories. You might want to create a subset of the dataset that includes only customers who are located in a particular city or customers who fall within a specific age range. By applying these criteria, you can extract a subset of the original dataset that contains only the relevant information, making it easier to analyze or work with a smaller, more focused portion of the data.

**What is Hive**

Hive is an open-source data warehousing and query execution framework built on top of Apache Hadoop. It provides a high-level interface, called HiveQL (Hive Query Language), which allows users to write SQL-like queries to analyse and process large volumes of structured and semi-structured data stored in Hadoop Distributed File System (HDFS) or other compatible file systems.

Hive simplifies the process of working with big data by providing a familiar SQL-like syntax that enables users to perform data querying, transformation, and analysis without the need for extensive programming knowledge. It translates HiveQL queries into a series of MapReduce jobs, which are then executed on the Hadoop cluster.

**What is YARN ?**

YARN, which stands for Yet Another Resource Negotiator, is a core component of Apache Hadoop. It is a cluster management and resource allocation framework that separates the responsibilities of resource management and job scheduling from the Hadoop Distributed File System (HDFS) and MapReduce processing engine.

YARN serves as a platform for executing and managing distributed applications in a Hadoop cluster. It provides a centralized resource management system that efficiently allocates resources, such as CPU, memory, and disk, to different applications running on the cluster. This allows for efficient utilization of cluster resources and improved scalability.