**Ans : Components of Hadoop 2.x**

**Hadoop 2.x has the following three Major Components:**

* HDFS
* YARN
* MapReduce

**HDFS**

* HDFS stands for Hadoop Distributed File System. It is also know as HDFS V2 as it is part of Hadoop 2.x with some enhanced features. It is used as a Distributed Storage System in Hadoop Architecture.
* HDFS is the Hadoop file system and comprises two major components:

namespaces and blocks storage service.

* The namespace service manages operations on files and directories, such as creating and modifying files and directories. The block storage service implements data node cluster management, block operations and replication.
* In Hadoop 1, a single Namenode managed the entire namespace for a Hadoop cluster. With HDFS federation, multiple Namenode servers manage namespaces and this allows for horizontal scaling, performance improvements, and multiple namespaces.
* The implementation of HDFS federation allows existing Namenode configurations to run without changes.
* For Hadoop administrators, moving to HDFS federation requires formatting Namenodes, updating to use the latest Hadoop cluster software, and adding additional Namenodes to the cluster.
* **Hadoop Distributed File System** is a block-structured file system where each file is divided into blocks of a pre-determined size. These blocks are stored across a cluster of one or several machines.
* Apache Hadoop HDFS Architecture follows a Master/Slave Architecture, where a cluster comprises of a single NameNode (Master node) and all the other nodes are DataNodes (Slave nodes).HDFS can be deployed on a broad spectrum of machines that support Java. Though one can run several DataNodes on a single machine, but in the practical world, these DataNodes are spread across various machines.
* The default size of each block is 128 MB in Apache Hadoop 2.x (64 MB in Apache Hadoop 1.x) which you can configure as per your requirement.

**YARN**

* HDFS federation brings important measures of scalability and reliability to Hadoop. YARN, the other major advance in Hadoop 2, brings significant performance improvements for some applications, supports additional processing models, and implements a more flexible execution engine.
* YARN is a resource manager that was created by separating the processing engine and resource management capabilities of MapReduce as it was implemented in Hadoop 1. YARN is often called the operating system of Hadoop because it is responsible for managing and monitoring workloads, maintaining a multi-tenant environment, implementing security controls, and managing high availability features of Hadoop.
* Like an operating system on a server, YARN is designed to allow multiple, diverse user applications to run on a multi-tenant platform. In Hadoop 1, users had the option of writing MapReduce programs in Java, in Python, Ruby or other scripting languages using streaming, or using Pig, a data transformation language. Regardless of which method was used, all fundamentally relied on the MapReduce processing model to run.
* YARN supports multiple processing models in addition to MapReduce. One of the most significant benefits of this is that we are no longer limited to working the often I/O intensive, high latency MapReduce framework. This advance means Hadoop users should be familiar with the pros and cons of the new processing models and understand when to apply them to particular use cases.

**MapReduce**

* MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.
* The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes.
* MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
* **Map stage** : The map or mapper’s job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
* **Reduce stage** : This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
* During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
* The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.