**List the components of Hadoop 2.x and explain them in detail.**

Hadoop 2.x has 3 major components-

1. HDFS

2. Map Reduce

3. YARN

**HDFS**- HDFS is Hadoop distributed File System which is responsible for the storage part of Big Data. HDFS manages the storage of large data files by dividing the disk in blocks of a fixed size. HDFS provides storage and access to data across Hadoop clusters. Like Hadoop related technologies HDFS is a key tool to manage pools of big data and supporting big data analytics applications.

* HDFS provides scalable,fault tolerant, storage at low cost architecture.
* HDFS software detects and compensates for hardware issues including disk and server failure.
* HDFS stores file across the collection of servers in a cluster.
* Files are decomposed in blocks, written to more than one of servers.
* Replication provides fault tolerance and performance.
* HDFS divides the disk in fixed size of blocks. Default size of a block is 128 MB in Hadoop 2.x.
* HDFS stores data in contigiuos blocks and each of the block get loaded on different machines.
* These blocks are replicated many times for providing redundancy and fault gtolerance to the data saved. Default value for replication is 3.
* HDFS in Hadoop 2.x have 2 daemons for HDFS- Master Daemon which is called Name Node Slave Daemon which is called Data Node.

**Map Reduce**- Map Reduce is responsible for the processing the big data on Hadoop. Map Reduce is a java based system created by Google. Map Reduce breaks down a big data processing job into smaller tasks to reduce the complexity of processing. Map reduce hav two functions Map function and Reduce function, Map function sends the processing code to all the commodity machines connected with Hadoop and this map function at all the commodity machines runs individually to process the data. After processing the big data , result is collected by reduce function at the high end machine which is our server and then the result is merged from all the different commodity machines. Hadoop map reduce architecture supports parallel processing. MapReduce architecture is based on YARN architecture. Hadoop Map Reduce has following Daemons- -Master Daemon which was called Job Tracker in Hadoop 1.x but called Resource Manager in Hadoop 2.x. -Similarly Slave Daemon which was called Task Tracker in Hadoop 1.x but called Node Manager in Hadoop 2.x. MapReduce framework forms the computer node while the HDFS file system forms the data node.

**YARN**- YARN is Yet Another Resource Negotiator its fundamental task is to split the functionalities of resource management and job scheduling into separate daemons. It extends the power of Hadoop so that Hadoop can become cost effective, linear scale storage and procesing system. Key Benefits of YARN are-

* + It offers improved cluster utilization.
  + It is Highly scalable
  + It is something Beyond Java
  + It provides novel programming models and sersvices
  + It is agile.

**Resource node**: Resource node has Resource Manager,which is a Per-Cluster Level Component. Resource Manager is again divided into two component scheduler and apllication manager. -Responsible to scheduler required resources to Applications (that is Per-Application Master). -It does only scheduling. -It does care about monitoring or tracking of those Applications

**Application Manager**- Application Master is a per-application level component.

* It Manages Life cycle. -It interacts with both Resource Manager’s Scheduler and Node Manager -It interacts with Node Manager to execute assigned tasks and monitor those task’s status.

**Node Manager**: Node Manager is a Per-Node Level component. -Managing the life-cycle of the Container. -Monitoring each Container’s Resources utilization.

**Container**: Each Master Node or Slave Node contains set of Containers.  
-Main Node Name contains a set of Containers. -Container is a portion of Memory. Container is similar to Data Slots in Hadoop 1.x

**fsimage** – An fsimage file contains the complete state of the file system at a point in time. Every file system modification is assigned a unique, monotonically increasing transaction ID. An fsimage file represents the file system state after all modifications up to a specific transaction ID.

**editslogs** – An edits file is a log that lists each file system change (file creation, deletion or modification) that was made after the most recent fsimage.