**Components of Hadoop 2.x:**

Hadoop2 Architecture has mainly 2 set of daemons

1) **HDFS 2.x Daemons -** Name Node, Secondary Name Node (not required in HA) and Data Nodes

* **HDFS High Availability (HA):** In Hadoop 1.x architecture Name Node was a single point of failure, which means if Name Node daemon is down somehow, we don’t have access to Hadoop Cluster than after.

To handle this problem in Hadoop 2.x supports two Name Nodes at a time one node is active and another is standby node, active Name Node handles the client operations in the cluster, Standby Name Node manages metadata same as Secondary Name Node in Hadoop 1.x When Active Name Node is down, Standby Name Node takes over and will handle the client operations then after.

* **HDFS Federation:** HDFS uses namespaces for managing directories, file and block level information in cluster. Hadoop 1.x architecture was able to manage only single namespace in a whole cluster with the help of the Name Node (which is a single point of failure in Hadoop 1.x). Once that Name Node is down we loose access of full cluster data.

To handle this problem Hadoop 2.x Architecture allows managing multiple namespaces by enabling multiple Name Nodes. So on HDFS shell we have multiple directories available but it may be possible that two different directories are managed by two active Name Nodes at a time. HDFS Federation by default allows single Name Node to manage full cluster (same as in Hadoop 1.x)

2) **MapReduce 2.x Daemons (YARN) -** Resource Manager, Node Manager

* MapReduce2 has replaced old daemon process Job Tracker and Task Tracker with YARN components Resource Manager and Node Manager respectively. These two components are responsible for executing distributed data computation jobs in Hadoop 2.x
* **Resource Manager:** This daemon process runs on master node (may run on the same machine as name node for smaller clusters) it is responsible for getting job submitted from client and schedule it on cluster, monitoring running jobs on cluster and allocating proper resources on the slave node. It communicates with Node Manager Daemon process on the slave node to track the resource utilization. It uses two other processes named Application Manager and Scheduler for MapReduce task and resource management
* **Node Manager:** This daemon process runs on slave nodes (normally on HDFS Data node machines) It is responsible for coordinating with Resource Manager for task scheduling and tracking the resource utilization on the slave node It also reports the resource utilization back to the Resource Manager It uses other daemon process like Application Master and Container for MapReduce task scheduling and execution on the slave node.