1. **Name node** 
   * NameNode is the centerpiece of HDFS also known as the Master Node
   * NameNode only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.
   * NameNode does not store the actual data or the dataset. The data itself is actually stored in the DataNodes.
   * NameNode knows the list of the blocks and its location for any given file in HDFS. With this information NameNode knows how to construct the file from blocks.
   * NameNode is so critical to HDFS and when the NameNode is down, HDFS/Hadoop cluster is inaccessible and considered down.
   * NameNode is a single point of failure in Hadoop cluster.
   * NameNode is usually configured with a lot of memory (RAM). Because the block locations are help in main memory.
2. **Data node**

* DataNode is responsible for storing the actual data in HDFS.
* DataNode is also known as the Slave
* NameNode and DataNode are in constant communication.
* When a DataNode starts up it announce itself to the NameNode along with the list of blocks it is responsible for.
* When a DataNode is down, it does not affect the availability of data or the cluster. NameNode will arrange for replication for the blocks managed by the DataNode that is not available.
* DataNode is usually configured with a lot of hard disk space. Because the actual data is stored in the DataNode.

1. **Resource Manager**
   * **Resource Manager** is the master.
   * It knows where the slaves are located (Rack Awareness) and how many resources they have.
   * It runs several services, the most important is the **Resource Scheduler** which decides how to assign the resources
2. **Node manager**
   * Node Manager(many per cluster) is the slave of the infrastructure.
   * When it starts, it announces himself to the Resource Manager.
   * Periodically, it sends an heartbeat to the Resource Manager.
   * Each Node Manager offers some resources to the cluster.
   * Its resource capacity is the amount of memory and the number of vcores. At run-time, the Resource Scheduler will decide how to use this capacity: a Container is a fraction of the NM capacity and it is used by the client for running a program