**Name node**

1. NameNode is the centerpiece of  HDFS, also known as the Master
2. 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.
3. 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.
4. 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.
5. NameNode is usually configured with a lot of memory (RAM). Because the block locations are help in main memory.

**Data node :**

1. DataNode is responsible for storing the actual data in HDFS. DataNode is also known as the Slave
2. NameNode and DataNode are in constant communication.
3. When a DataNode starts up it announce itself to the NameNode along with the list of blocks it is responsible for.
4. 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.
5. DataNode is usually configured with a lot of hard disk space. Because the actual data is stored in the DataNode.

**Resource Manager:**

1. Manages job scheduling and execution
2. Global resource allocation.
3. Containers to attempt resource restriction.
4. Resource managers do understand random very important requests and will move them to the Top, pushing all other requests to the side, understanding true priority without permanently reserving resources.
5. It is allocated one per cluster and it is considered the master. It knows where the nodes are located and how much resources do they hold. It runs a lot of services, most important one being the Resource Scheduler, which is in charge of assigning the resources.

**Node manager :**

1. Each slave node in Yet Another Resource Negotiator (YARN) has a Node Manager daemon, which acts as a slave for the Resource Manager
2. each slave node has a service that ties it to the processing service (Node Manager) and the storage service (DataNode) that enable Hadoop to be a distributed system.
3. Each Node Manager tracks the available data processing resources on its slave node and sends regular reports to the Resource Manager.
4. The processing resources in a Hadoop cluster are consumed in bite-size pieces called containers. A container is a collection of all the resources necessary to run an application: CPU cores, memory, network bandwidth, and disk space. A deployed container runs as an individual process on a slave node in a Hadoop cluster. All container processes running on a slave node are initially provisioned, monitored, and tracked by that slave node’s Node Manager daemon.
5. Node Manager oversees containers’ life-cycle management; monitoring resource usage (memory, CPU) of individual containers, tracking node-health, log’s management and auxiliary services which may be exploited by different YARN applications.