1. Explain the importance of below 4 demons in job execution with minimum of 5 points
   1. **NameNode**

* An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients.
* The NameNode executes file system namespace operations like opening, closing, and renaming files and directories.
* The NameNode stores the HDFS filesystem information in a file named fsimage. Updates to the file system (add/remove blocks) are not updating the fsimage file, but instead are logged into a file, so the I/O is fast append only streaming as opposed to random file writes. When restarting, the namenode reads the fsimage and then applies all the changes from the log file to bring the filesystem state up to date in memory.
* NameNode does not store the actual data or the dataset. The data itself is actually stored in the DataNodes. It only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.
* NameNode determines the mapping of blocks to DataNodes. It 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.
  1. **DataNode**
* The DataNode is a commodity hardware having the GNU/Linux operating system and DataNode software. For every node (Commodity hardware/System) in a cluster, there will be a DataNode. These nodes manage the data storage of their system.
* It is also known as a slave.
* In HDFS there are a number of DataNodes. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes.
* 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.
* The DataNodes are responsible for serving read and write requests from the file system’s clients.
* The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.
  1. **Resource Manager**
* The ResourceManager is the master.
* The ResourceManager is responsible for tracking the resources in a cluster (it knows knows where the slaves are located (Rack Awareness)), and scheduling applications (e.g. MapReduce jobs). Prior to Hadoop 2.4, the ResourceManager was the single point of failure in a YARN cluster. The High Availability feature adds redundancy in the form of an Active/Standby ResourceManager pair to remove this otherwise single point of failure.
* ResourceManager runs several services; the most important is the Resource Scheduler which decides how to assign the resources.
* ResourceManager is able to address the most important design requirements – scalability, support for alternate programming paradigms.
  1. **NodeManager**
* NodeManager (many per cluster) is the slave.
* The NodeManager is YARN’s per-node agent, and takes care of the individual compute nodes in a Hadoop cluster. This includes keeping up-to date with the ResourceManager (RM), overseeing 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.
* NodeManager takes instructions from the ResourceManager and manage resources available on a single node.
* When NodeManager starts, it announces himself to the Resource Manager. Periodically, it sends a heartbeat to the Resource Manager.
* Each NodeManager 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.