Namenode

1. NameNode is the centerpiece of HDFS.
2. NameNode is also known as the Master
3. NameNode only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.
4. NameNode does not store the actual data or the dataset. The data itself is actually stored in the DataNodes.
5. 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.
6. NameNode is so critical to HDFS and when the NameNode is down, HDFS/Hadoop cluster is inaccessible and considered down.
7. NameNode is a single point of failure in Hadoop cluster.
8. NameNode is usually configured with a lot of memory (RAM). Because the block locations are help in main memory.

DataNode

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

Resource Manager

1. **ResourceManager (RM)** is the master that arbitrates all the available cluster resources and thus helps manage the distributed applications running on the YARN system.
2. It works together with the per-node NodeManagers (NMs) and the per-application ApplicationMasters (AMs).
3. Responsible for maintaining a collection of submitted applications. Also keeps a cache of completed applications so as to serve users' requests via web UI or command line long after the applications in question finished.
4. RM needs to gate the user facing APIs like the client and admin requests to be accessible only to authorized users. This component maintains the ACLs lists per application and enforces them whenever a request like killing an application, viewing an application status is received.
5. Maintains a thread-pool to launch AMs of newly submitted applications as well as applications whose previous AM attempts exited due to some reason. Also responsible for cleaning up the AM when an application has finished normally or forcefully terminated.
6. The Scheduler is responsible for allocating resources to the various running applications subject to constraints of capacities, queues etc. It performs its scheduling function based on the resource

requirements of the applications such as memory, CPU, disk, network etc. Currently, only memory is supported and support for CPU is close to completion.

Node Manager

1. On startup, this component registers with the RM and sends information about the resources available on the nodes.
2. Subsequent NM-RM communication is to provide updates on container statuses – new containers running on the node, completed containers, etc
3. NM accepts requests from Application Masters (AMs) to start new containers, or to stop running ones.
4. The NM provides a framework for extending its functionality by configuring auxiliary services. This allows per-node custom services that specific frameworks may require, and still sandbox them from the rest of the NM.
5. Interacts with the underlying operating system to securely place files and directories needed by containers and subsequently to launch and clean up processes corresponding to containers in a secure manner.
6. Provides functionality of checking the health of the node by running a configured script frequently. It also monitors the health of the disks specifically by creating temporary files on the disks every so often. Any changes in the health of the system are notified to NodeStatusUpdater (described above) which in turn passes on the information to the RM.
7. Exposes the list of applications, containers running on the node at a given point of time, node-health related information and the logs produced by the containers.