Answers

**Name node**

1. The NameNode is the centerpiece of an HDFS file system.
2. It keeps the directory tree of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these files itself.
3. Client applications talk to the NameNode whenever they wish to locate a file, or when they want to add/copy/move/delete a file. The NameNode responds the successful requests by returning a list of relevant [DataNode](https://wiki.apache.org/hadoop/DataNode) servers where the data lives.
4. The NameNode is a [Single Point of Failure](https://wiki.apache.org/hadoop/Single%20Point%20of%20Failure) for the HDFS Cluster. HDFS is not currently a High Availability system. When the NameNode goes down, the file system goes offline.
5. The namenode stores the HDFS filesystem information in a file named fsimage.

**Data node**

1. A DataNode stores data in the [Hadoop File System](https://wiki.apache.org/hadoop/HadoopFileSystem).
2. A functional filesystem has more than one DataNode, with data replicated across them.
3. On startup, a DataNode connects to the NameNode, spinning until that service comes up. It then responds to requests from the [NameNode](https://wiki.apache.org/hadoop/NameNode) for filesystem operations.
4. Client applications can talk directly to a DataNode, once the [NameNode](https://wiki.apache.org/hadoop/NameNode) has provided the location of the data.
5. DataNode instances can talk to each other, which is what they do when they are replicating data.

**Resource Manager**

1. ResourceManager is the master that arbitrates all the available cluster resources and thus helps manage the distributed applications running on the YARN system.
2. ResourceManager is primarily limited to scheduling i.e. only arbitrating available resources in the system among the competing applications and not concerning itself with per-application state management.
3. The ResourceManager (RM) is responsible for tracking the resources in a cluster, and scheduling applications (e.g., MapReduce jobs).
4. ResourceManager is realized through an Active/Standby architecture - at any point of time, one of the RMs is Active, and one or more RMs are in Standby mode waiting to take over should anything happen to the Active.

**Node manager**

1. The **Node Manager** (many per cluster) is the slave of the infrastructure. When it starts, it announces himself to the Resource Manager.
2. Each Node Manager offers some resources to the cluster. Its resource capacity is the amount of memory and the number of vcores.
3. Each Node Manager tracks the available data processing resources on its slave node and sends regular reports to the Resource Manager.