Hadoop

- What are the differences between regular FileSystem and HDFS?

  Ans - A regular file system stores data on a single machine and is not fault tolerant, whereas HDFS is a distributed file system designed to store large datasets across multiple nodes with replication for fault tolerance and high throughput processing. In regular Filesystem data is stored on disk while HDFS stores metadata.

- What is HDFS fault-tolerant?

Ans:HDFS fault- tolerant means the system continues working even if some machine fails and data is not lost when node crashes. As it stores multiple copies of each data across different DataNodes. If one node fails, data can be accessed from another replica and the system automatically recreates lost replicas.

- What are the two types of metadata that a NameNode server holds

Ans: NameNode stores two types of metadata: first, the file system namespace metadata such as directory structure, permissions, and file-to-block mapping; second, block location metadata which tracks where each block is stored across DataNodes.

- If you have an input file of 350 MB, how many blocks would HDFS create and what would be the size of each block? Assuming default block size is 128MB.

  Ans: file size = 350 mb
  Default size = 128 mb
  HDFS will create 3 block , block 1 -128 mb, block 2 - 128 mb and block 3 -94 mb.

- How does rack awareness work in HDFS?

Ans- Rack awareness in HDFS is a replica placement strategy where the NameNode distributes block replicas across different racks to improve fault tolerance and reduce network congestion. Typically, one replica is placed locally, and others are placed on different racks.

- Is it possible to change the number of mappers to be created in a MapReduce job?

  Ans: In MapReduce, we cannot directly control the number of mappers. The number of mappers is determined by the number of input splits, which depends on the file size and HDFS block size. However, we can indirectly influence it by configuring split size.

- What is the distributed cache in MapReduce?

  Ans: Distributed Cache is a feature in Mapreduce that distributes small read-only file to all DataNodes before the job starts so every task can access the file locally.