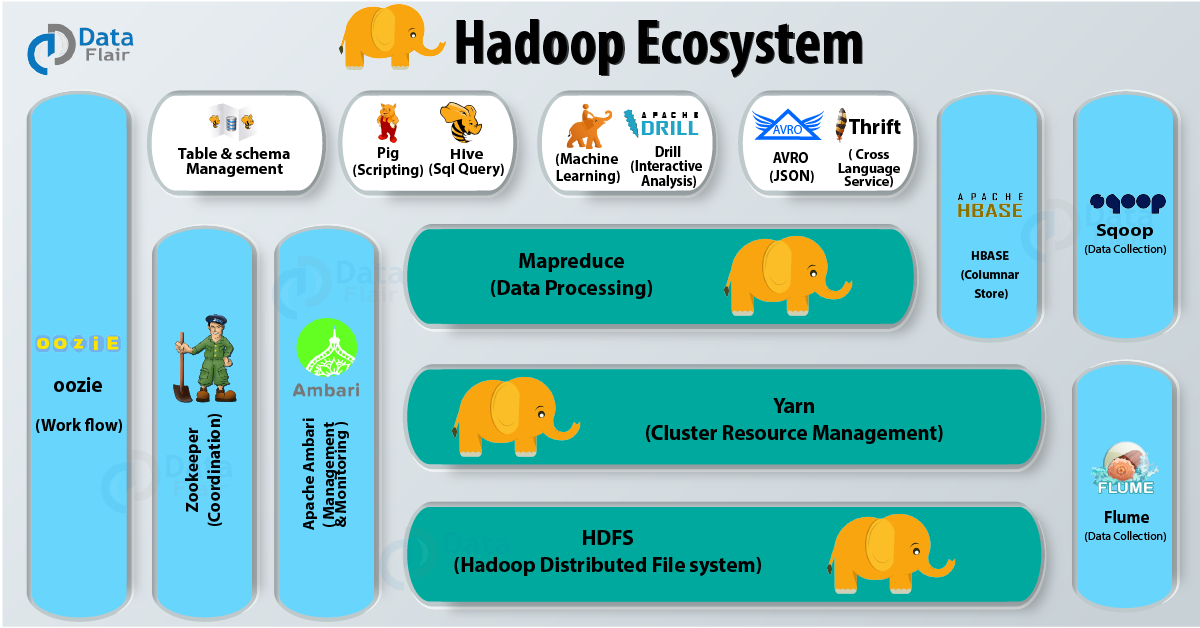
**Ecosystem**



Hadoop Ecosystem Components are

🡪HDFS

🡪Yarn

🡪MapReduce

🡪Hive

🡪Pig

🡪Hbase (Apache HBASE)

🡪MahOut

🡪Zookeeper

🡪Oozie

🡪Sqoop

🡪Flume (also in combo of flume+kafta)

🡪Apache Drill

🡪Apache Spark

### 🡪Solr And Lucene

### HDFS

### 🡪Hadoop Distributed File System

### 🡪Storage layer of Hadoop

### 🡪It is Java software that provides many features like scalability, high availability, fault tolerance, cost effectiveness etc.

### 🡪We can deploy many other software frameworks over HDFS.

### Component of Hadoop Distributed File System (HDFS)

### 1.Name Node

### 2.Secondary Name Node

### 3.Data Node

### Hadoop Ecosystem

### Name Node

🡪NameNode is nothing but the master node. The NameNode is responsible for managing file system namespace, controlling the client’s access to files. Also, it executes tasks such as opening, closing and naming files and directories.

🡪 NameNode has two major files – FSImage and Edits log

**FSImage –** FSImage is a point-in-time snapshot of HDFS’s metadata. It contains information like file permission, disk quota, modification timestamp, access time etc.

**Edits log –** It contains modifications on FSImage. It records incremental changes like renaming the file, appending data to the file etc.

Whenever the NameNode starts it applies Edits log to FSImage. And the new FSImage gets loaded on the NameNode.

### Secondary Name Node

### 🡪If the NameNode has not restarted for months the size of Edits log increases. This, in turn, increases the downtime of the cluster on the restart of NameNode.

### 🡪In this case, Secondary NameNode comes into the picture. The Secondary NameNode applies edits log on FSImage at regular intervals. And it updates the new FSImage on primary NameNode.

### Data Node

These are the nodes which store the actual data. HDFS stores the data in a distributed manner. It divides the input files of varied formats into blocks. The Data nodes stores each of these blocks.

functions of Data nodes:-

🡪On start up, Data node does handshake with Name node. It verifies the namespace ID and software version of Data node.

🡪Also, it sends a block report to Name node and verifies the block replicas.

🡪It sends a heartbeat to Name node every 3 seconds to tell that it is alive.

**2.Map Reduce**

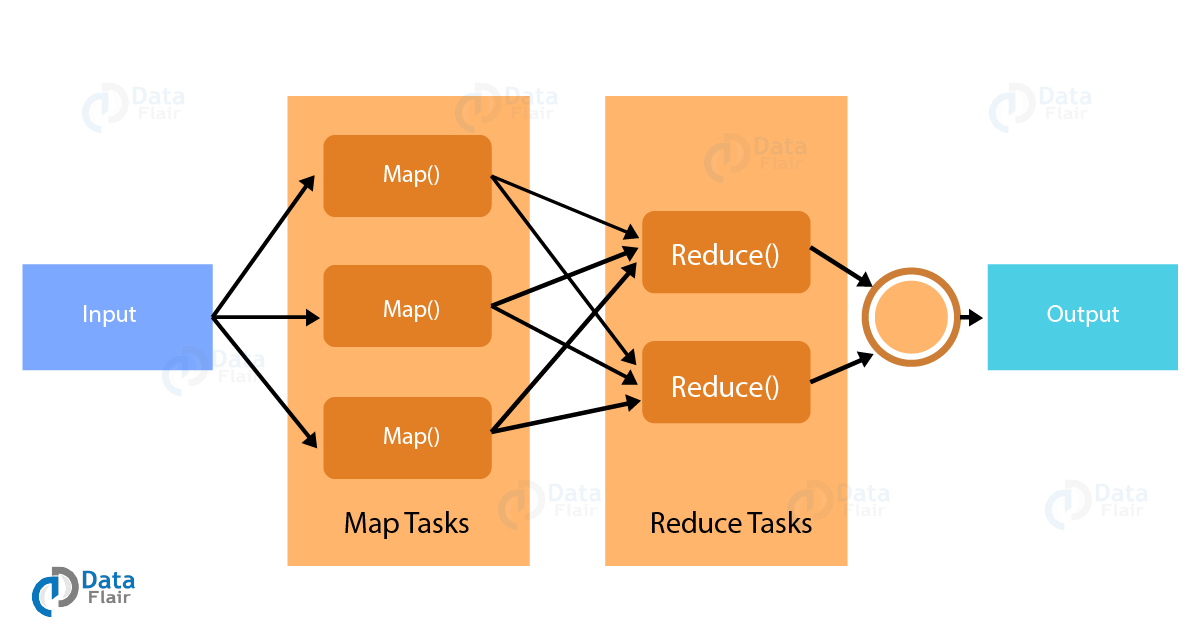
🡪Data Processing layer of Hadoop

🡪 It applies the computation on sets of data in parallel thereby improving the performance.

MapReduce works in two phases –

🡪Map Phase – This phase takes input as key-value pairs and produces output as key-value pairs. It can write custom business logic in this phase. Map phase processes the data and gives it to the next phase.

🡪Reduce Phase – The MapReduce framework sorts the key-value pair before giving the data to this phase. This phase applies the summary type of calculations to the key-value pairs.



🡪Mapper reads the block of data and converts it into key-value pairs.

🡪Now, these key-value pairs are input to the reducer.

🡪The reducer receives data tuples from multiple mappers.

🡪Reducer applies aggregation to these tuples based on the key.

🡪The final output from reducer gets written to HDFS.

**MapReduce framework takes care of the failure. It recovers data from another node in an event where one node goes down.**

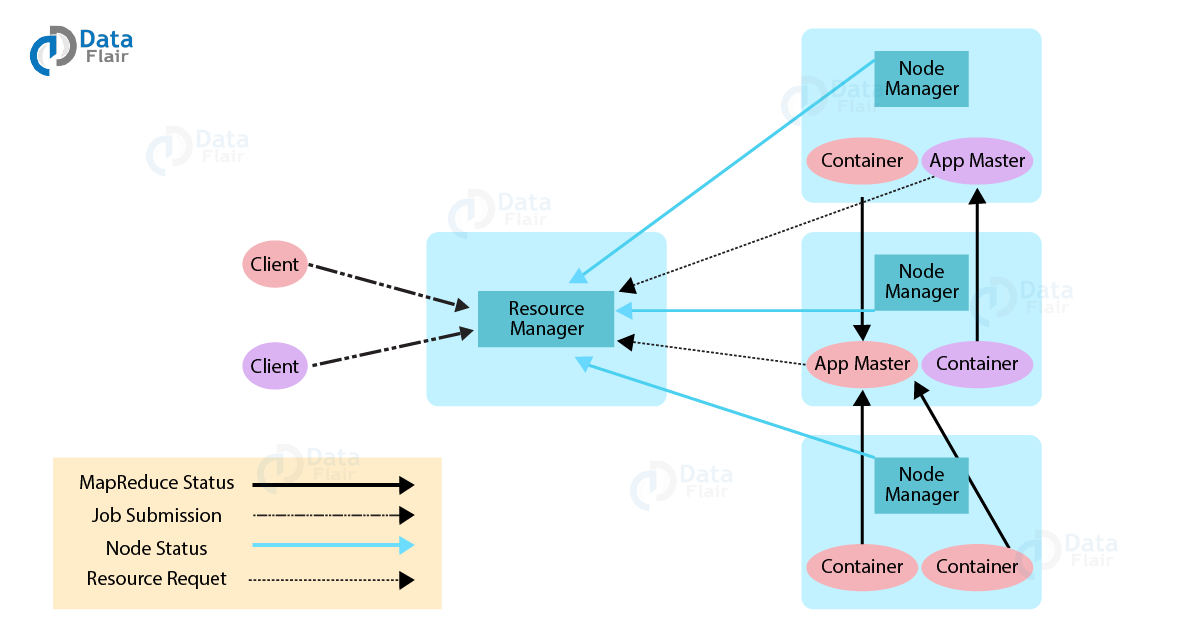
**3.Yarn**

🡪Yet Another Resource Manager.

🡪It is like the operating system of Hadoop as it monitors and manages the resources.

🡪Yarn came into the picture with the launch of Hadoop 2.x in order to allow different workloads.

🡪It handles the workloads like stream processing, interactive processing, batch processing over a single platform. Yarn has two main components – Node Manager and Resource Manager.



**4. Hive**

[Hive is a data warehouse project](https://data-flair.training/blogs/apache-hive-tutorial/) built on the top of Apache Hadoop which provides data query and analysis. It has got the language of its own call HQL or Hive Query Language. HQL automatically translates the queries into the corresponding map-reduce job.

Main parts of the Hive are –

🡪MetaStore – it stores metadata

🡪Driver – Manages the lifecycle of [HQL statement](https://data-flair.training/blogs/hiveql-select-statement/)

🡪Query compiler – Compiles HQL into DAG i.e. Directed Acyclic Graph

🡪Hive server – Provides interface for JDBC/ODBC server.



**5.Pig**

🡪Similar to Hive

🡪No use of this platform in the current market. Hive takes over pig .

🡪Created by Yahoo

**6. HBASE**

🡪HBase is a NoSQL database built on the top of HDFS. The various [**features of**HBase](https://data-flair.training/blogs/features-of-hbase/) are that it is open-source, non-relational, distributed database.

🡪It imitates **Google’s Bigtable** and written in Java.

Component:

🡪HBASE MASTER

🡪REGION SERVER

**7.Sqoop**

🡪[Sqoop imports data](https://data-flair.training/blogs/sqoop-import/) from external sources into compatible Hadoop Ecosystem components like HDFS, Hive, HBase etc.

🡪It also transfers data from Hadoop to other external sources. It works with RDBMS like TeraData, Oracle, MySQL and so on.

🡪The major difference between Sqoop and Flume is that Flume does not work with structured data. But Sqoop can deal with structured as well as unstructured data.

Let us see how Sqoop works

When we submit Sqoop command, at the back-end, it gets divided into a number of sub-tasks. These sub-tasks are nothing but map-tasks. Each map-task import a part of data to Hadoop. Hence all the map-task taken together imports the whole data.

[Sqoop export](https://data-flair.training/blogs/sqoop-export/) also works in a similar way. Only thing is instead of importing, the map-task export the part of data from Hadoop to destination database.

**8.Flume**

**🡪It is a pull system which does not affect the existing architecture.**

🡪It is a service which helps to ingest structured and semi-structured data into HDFS.

🡪[Flume](https://data-flair.training/blogs/apache-flume-tutorial/) works on the principle of distributed processing. It aids in collection, aggregation, and movement of a huge amount of data sets.

**9.Apache Spark**

🡪[Apache Spark](https://data-flair.training/blogs/spark-tutorial/) unifies all kinds of Big Data processing under one umbrella. It has built-in libraries for streaming, SQL, machine learning and graph processing.

🡪Apache Spark is lightening fast. It gives good performance for both batch and stream processing. It does this with the help of DAG scheduler, query optimizer, and physical execution engine.

🡪Spark offers 80 high-level operators which makes it easy to build parallel applications. Spark has various libraries like MLlib for machine learning, GraphX for graph processing, SQL and Data frames, and Spark Streaming. One can run Spark in standalone cluster mode on Hadoop, Mesos, or on Kubernetes.

🡪 One can write Spark applications using SQL, R, Python, Scala, and Java. As such Scala in the native language of Spark. It was originally developed at the University of California, Berkley. Spark does in-memory calculations. This makes [Spark faster than Hadoop map-reduce](https://data-flair.training/blogs/spark-vs-hadoop-mapreduce/).