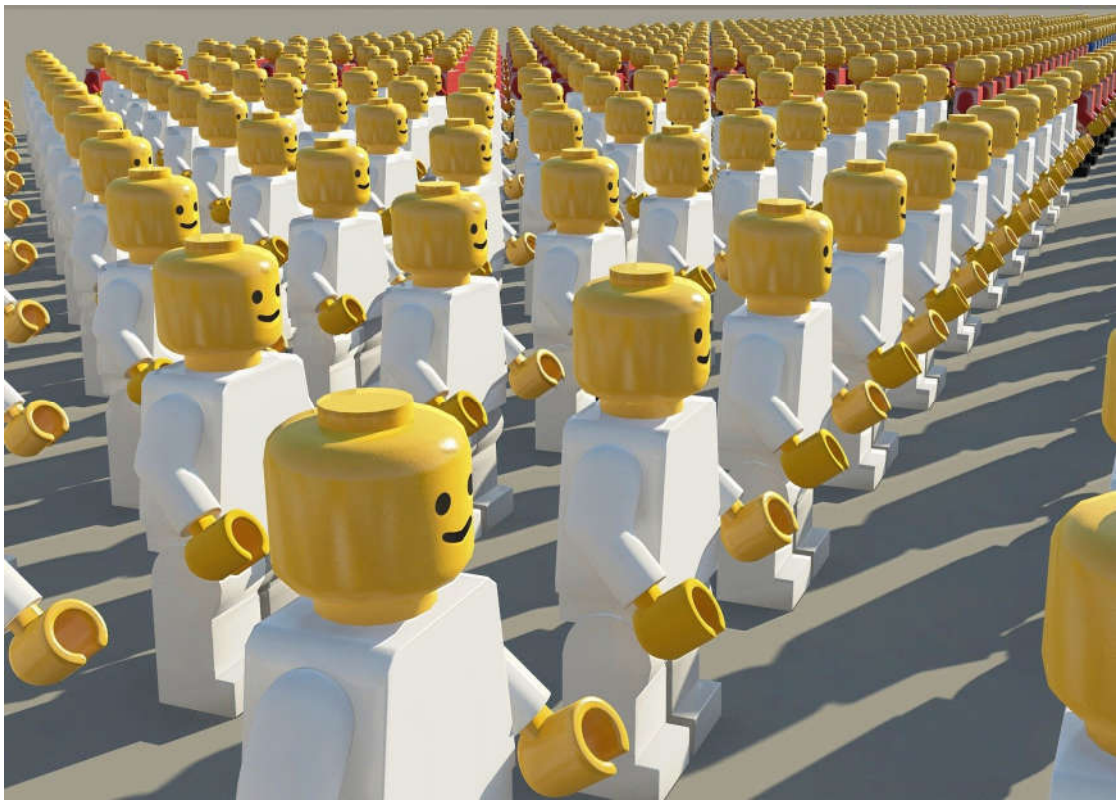
Introduction to Distributed File Systems

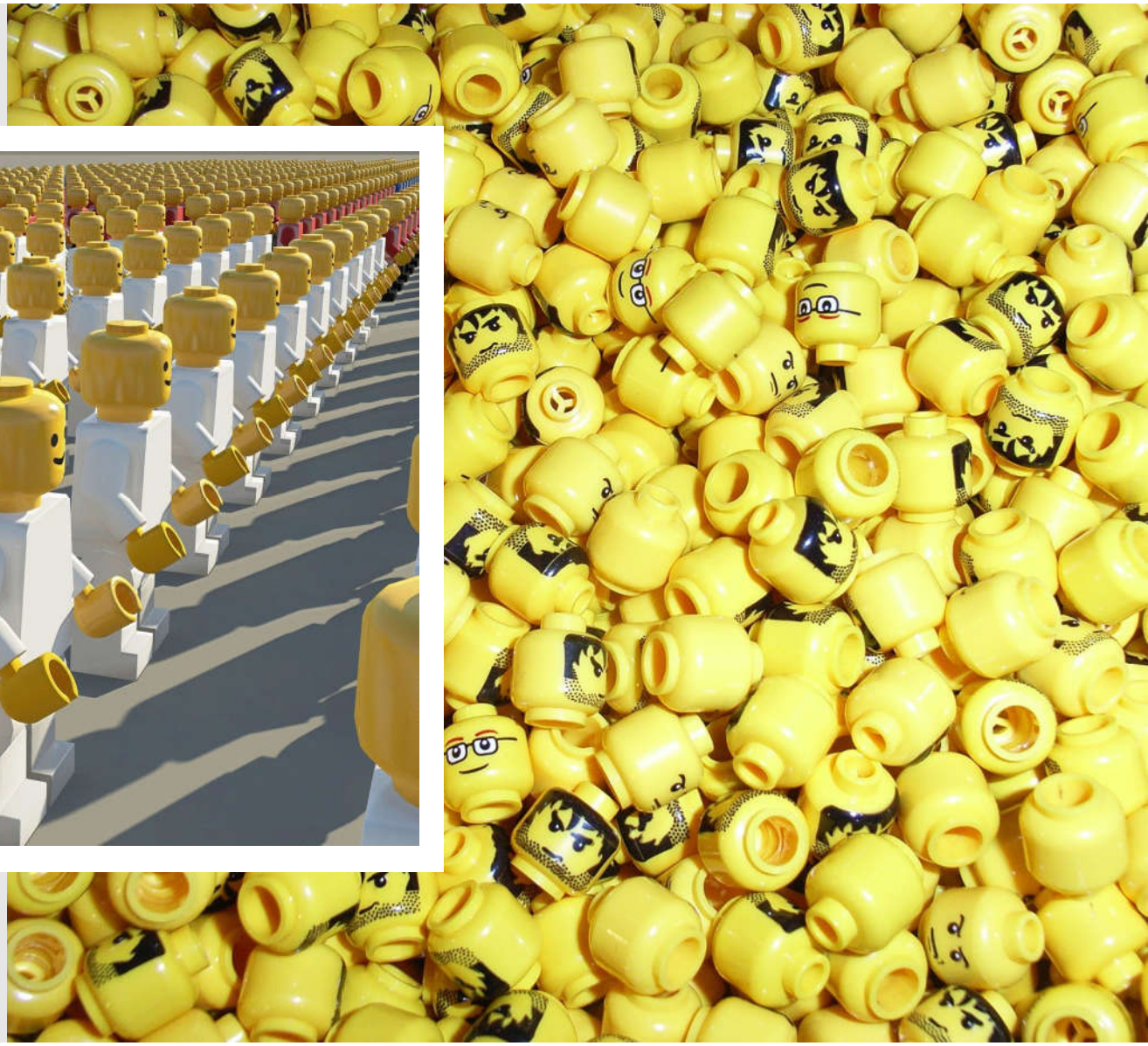
Harvesting, Storing, and Retrieving Data


How Much Data Do We Make?
How Much Data?

JUST HOW BIG IS BIG

1.7MB of data is created every second by every person during 2020.
In the last two years alone, an astonishing 90% of the world's data has been created.
2.5 quintillion bytes of data are produced by humans every day.
463 exabytes of data will be generated each day by humans as of 2025.
95 million photos and videos are shared every day on Instagram.
By the end of 2020, 44 zettabytes will make up the entire digital universe.
Every day, 306.4 billion emails are sent, and 5 million Tweets are made.

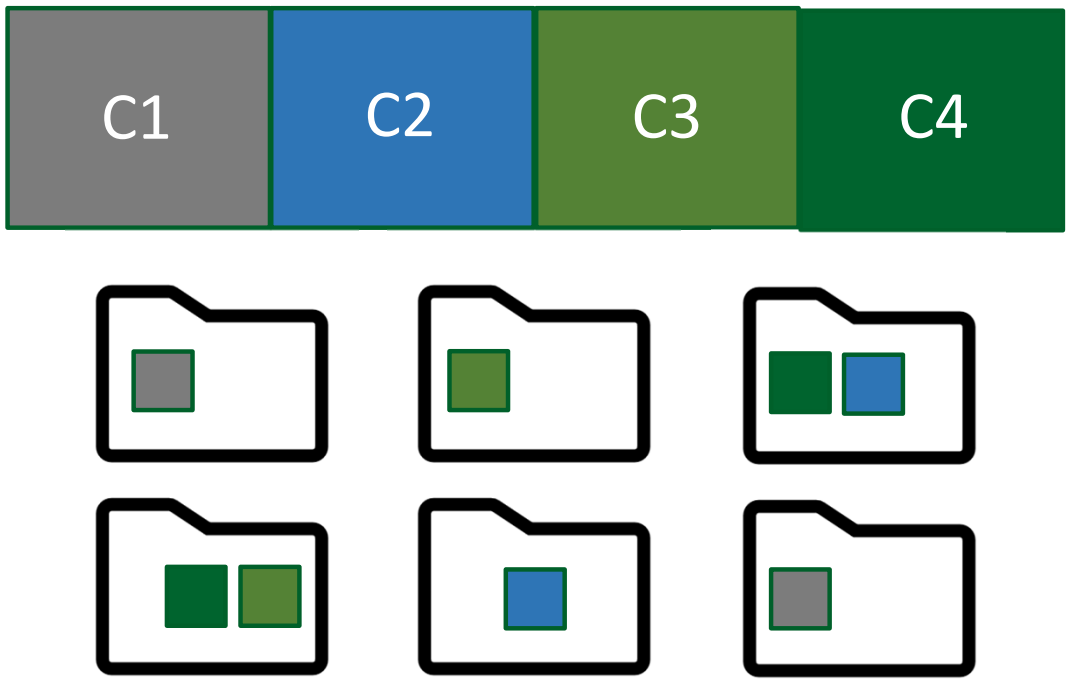




BIG DATA: HOW TO MANAGE IT?

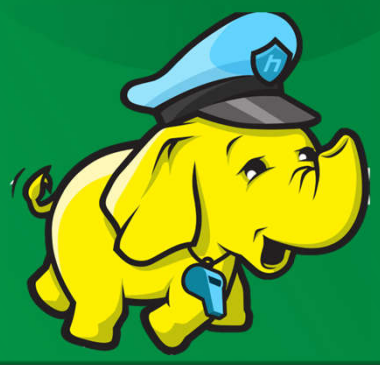
- Can we use the well-known relational database systems like Oracle, Microsoft, SQL Server, IBM DB2, MySQL, etc., to efficiently manage big data?
- NO!
- Because big data is very different from the data that can be found in relational database systems.
- In other words, big data is NOT the traditional data that is big.
- We need a new type of data management system!

Google File System



BIG DATA: HOW TO MANAGE IT?
Enter, Apache Hadoop Framework

APACHE HADOOP FRAMEWORK: A BIT OF HISTORY



- 2002: Doug Cutting & Mike Cafarella built Nutch to crawl the web. Nutch is a primitive distributed system that could run on several computers at most.
- 2003: Google released the white paper on Google File System that was also a distributed system but much more efficient and powerful.
- 2004: Google released the white paper on MapReduce that is a parallel processing system.
- Based on these white papers, Doug Cutting & Mike Cafarella built a new distributed system that is the foundation for an open-source distributed system: Apache Hadoop Framework.
- And History was made!

APACHE HADOOP FRAMEWORK: OVERVIEW

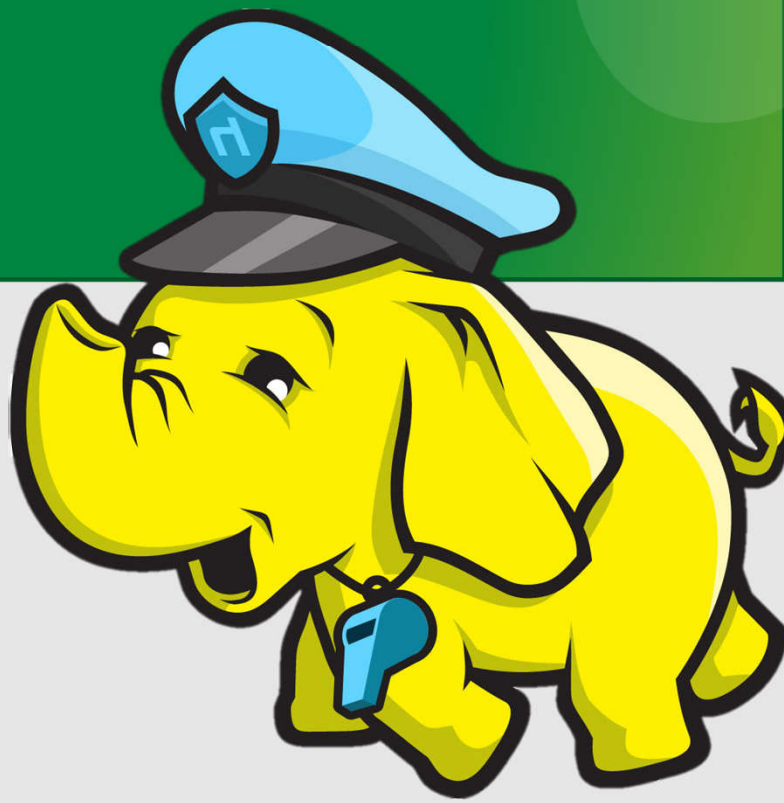
What is Apache Hadoop?

- Apache Hadoop is an open-source framework:
  - Capable of processing large amounts of heterogeneous data sets in a distributed fashion across clusters of commodity computers and hardware
  - Uses a simplified programming model
  - Provides a reliable shared storage and analysis system.

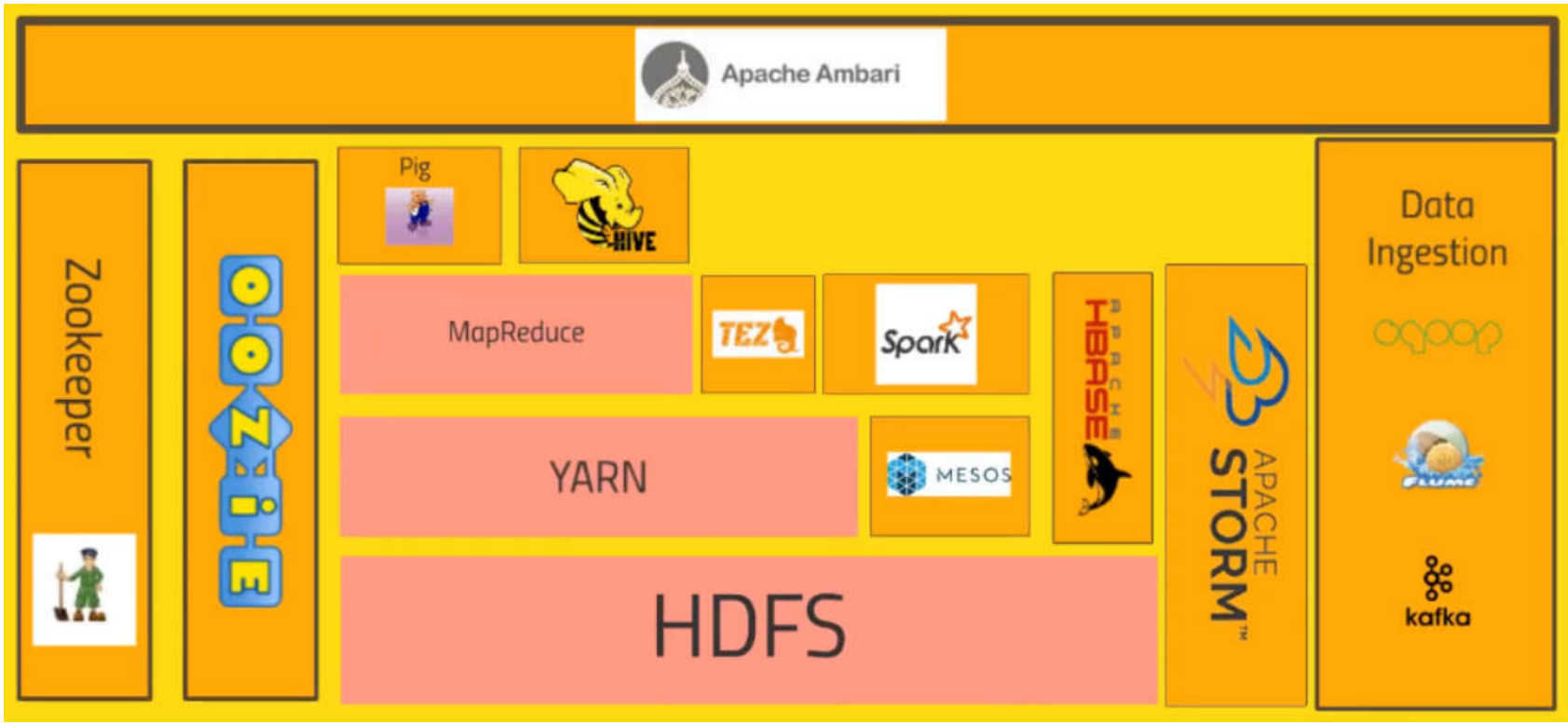

Hadoop Tutorial For Beginners | Hadoop Ecos...
Hadoop
Ecosystem
Explained in
20 min!
Frank Kane

APACHE HADOOP ECOSYSTEM: CORE COMPONENTS

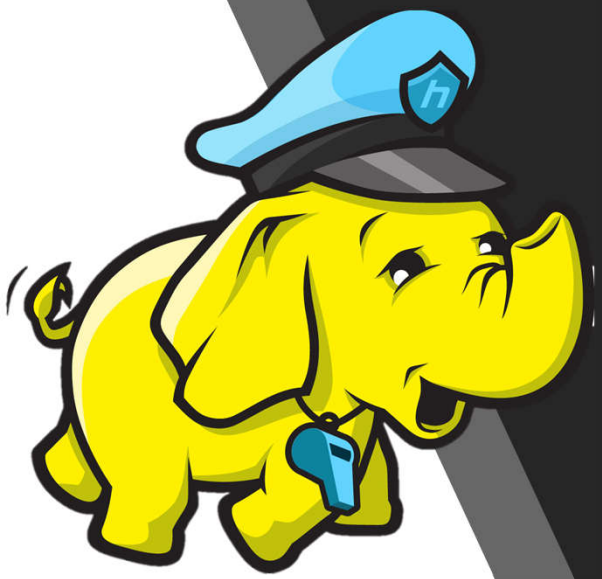
- Hadoop Distributed File System (HDFS)
- YARN (Hadoop 2)
- MapReduce
- Hive
- Spark



APACHE HADOOP ECOSYSTEM: CORE COMPONENTS

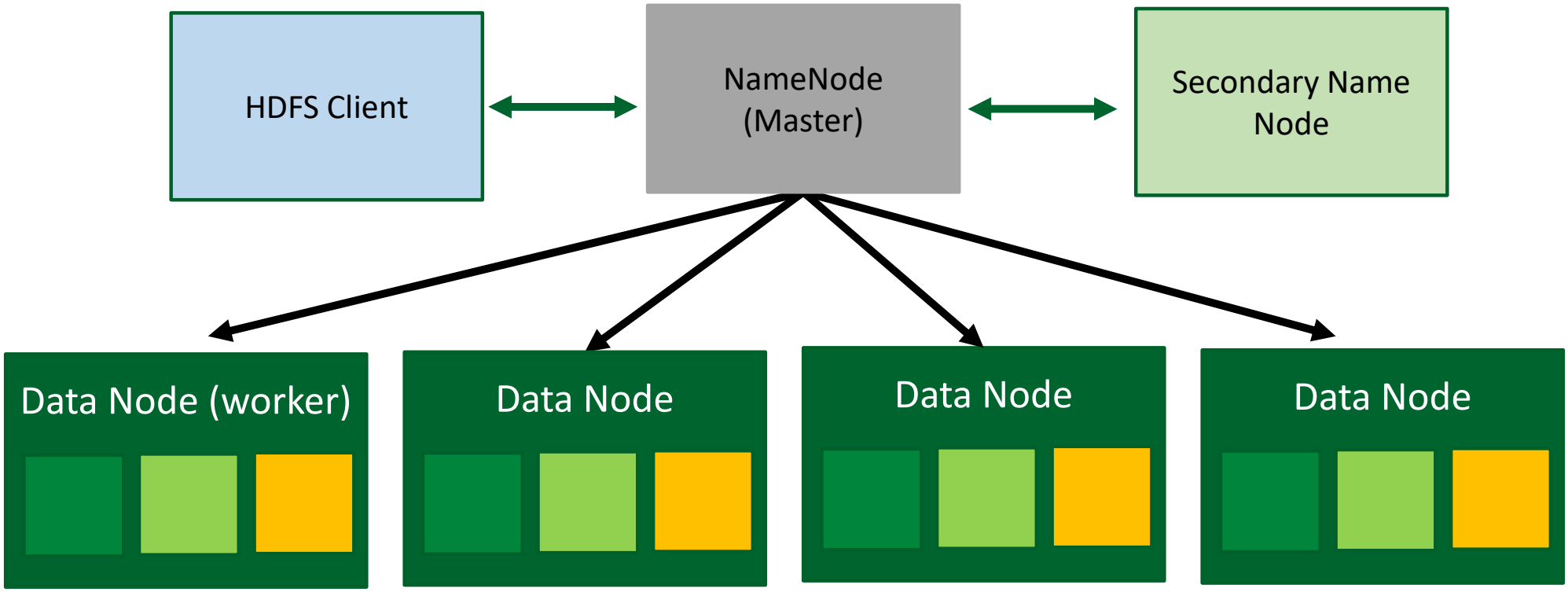


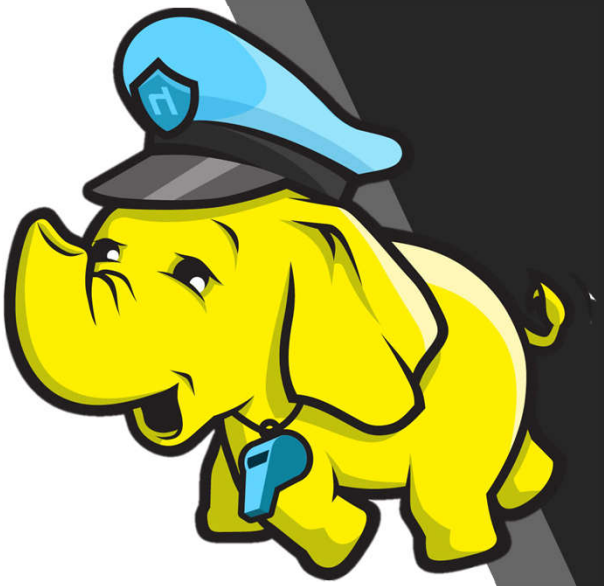
APACHE HADOOP ECOSYSTEM: HDFS



- The"secret sauce"
- Clusters of “computers”
- HDFS is comprised of 3 important components:
  - NameNode
  - DataNode
  - Secondary NameNode
- HDFS operates on a Master-Slave architecture model:
  - NameNode acts as the master node for keeping a track of the storage cluster.
  - DataNode acts as a slave node summing up to the various systems within a Hadoop Cluster

Apache Hadoop Ecosystem: HDFS Arichitecture

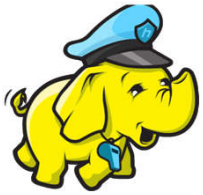




HADOOP ECOSYSTEM: Yet Another Resource Negotiator (YARN)

- Introduced in Hadoop2
- Split out from MapReduce, so that alternatives to MapReduce (such as Spark) could be built on top of YARN
- While HDFS splits up the data storage across your cluster, YARN splits up the computation
- YARN aligns worker nodes and storage nodes to run jobs as efficiently as possible
- The resource manager starts a MapReduce application. It first spawns an application master on some node that has capacity.
- The application master then requests additional nodes that run different application processes (which are usually Map or Reduce tasks).

APACHE HADOOP ECOSYSTEM: YARN

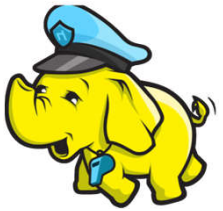


Hadoop v1.0

Map Reduce
Data processing & Resource Management

HDFS
Distributed File Storage

Hadoop v2.0



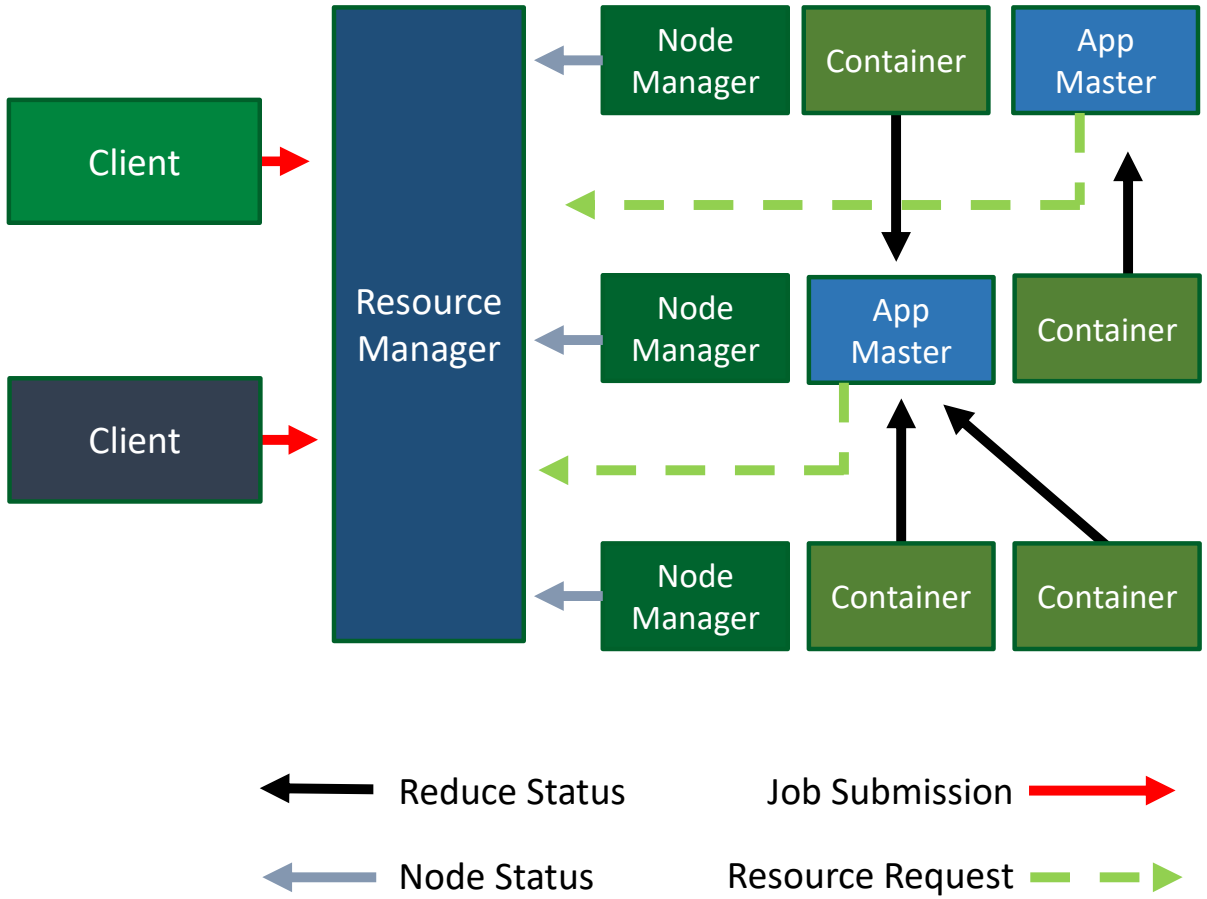
Map Reduce
Data Processing

Other
Data Processing Frameworks

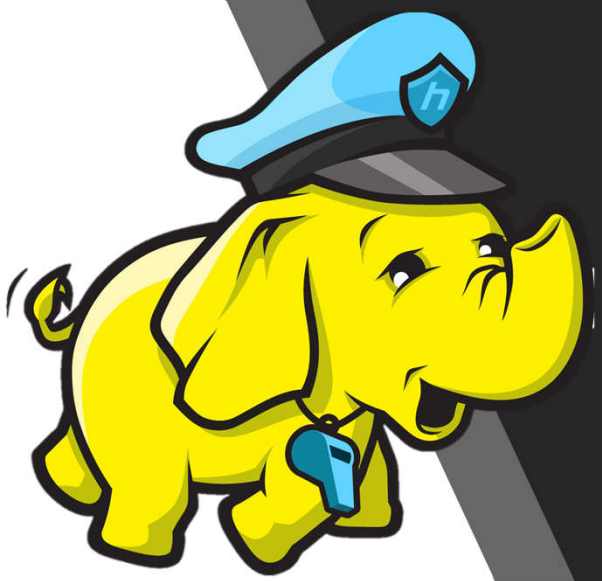
YARN
Resource Management

HDFS
Distributed File Storage

APACHE HADOOP ECOSYSTEM: YARN Architecture



Apache Hadoop Ecosystem: MapReduce



- MapReduce is where data from HDFS gets processed efficiently.
  - Breaks down a big data processing job into smaller tasks
  - Responsible for analyzing large datasets in parallel before finding the results.
- The basic principle of operation behind MapReduce:
  - “Map” job sends a query for processing to various nodes in the cluster.
  - “Reduce” job collects all the results to output into a single value.

Apache Hadoop Ecosystem: MapReduce

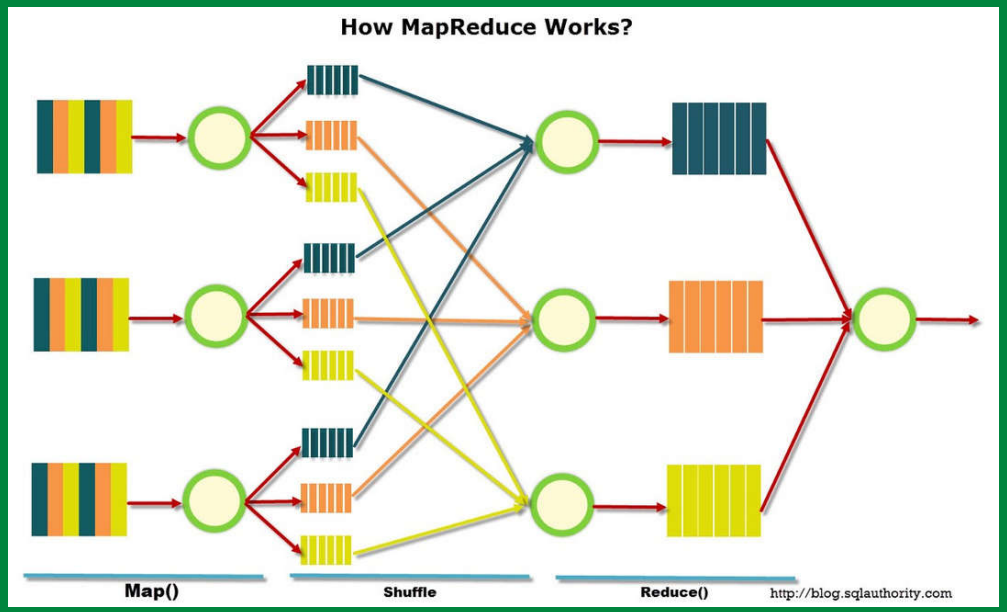


Image Source

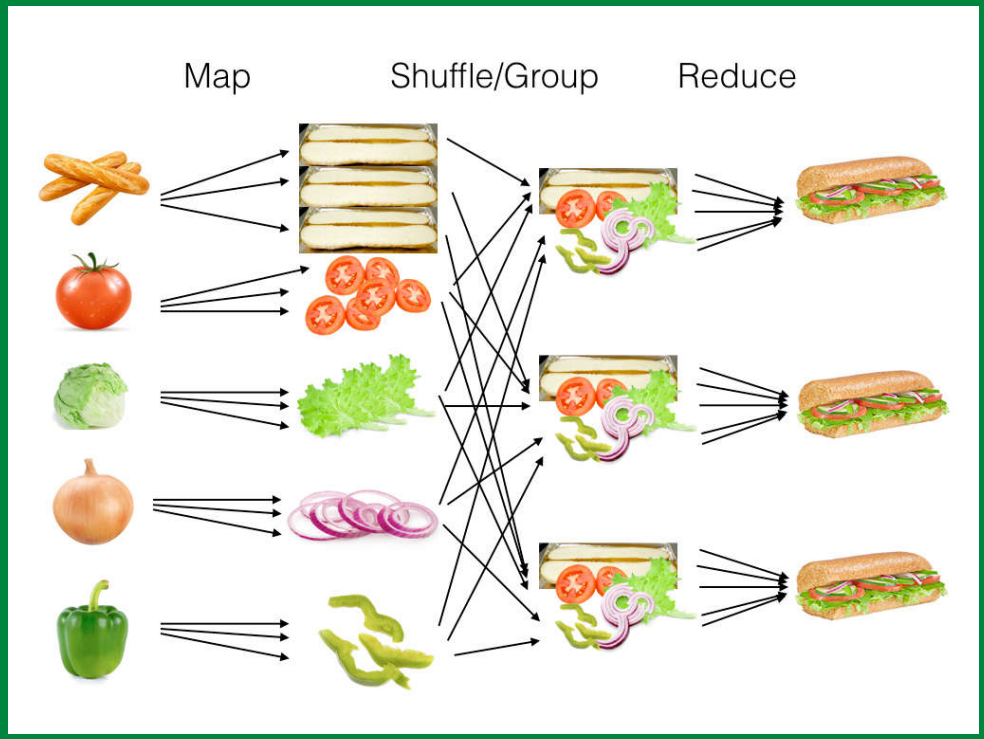
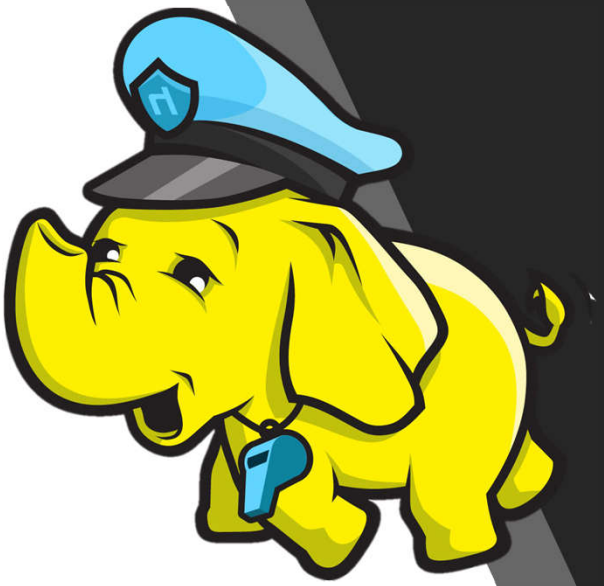


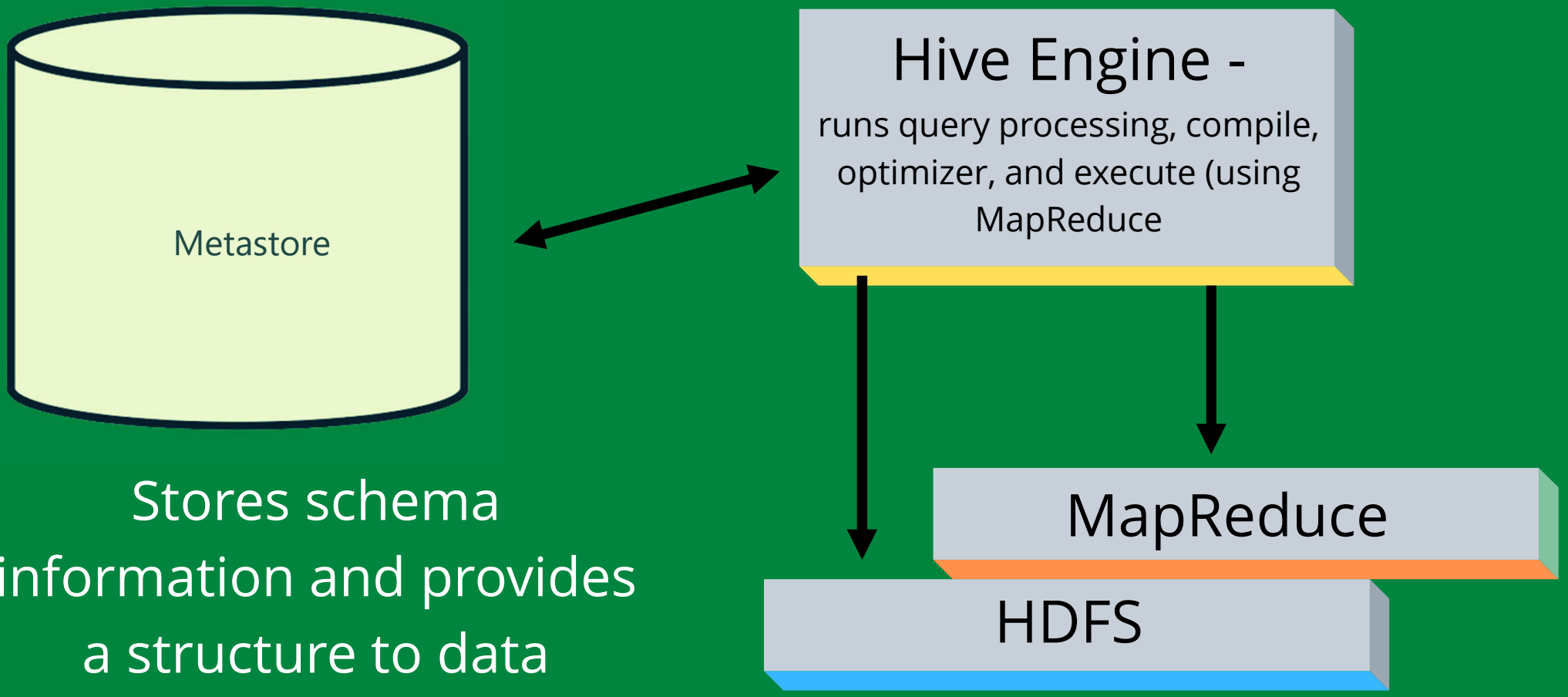
Image Source

Apache Hadoop Ecosystem: Hive



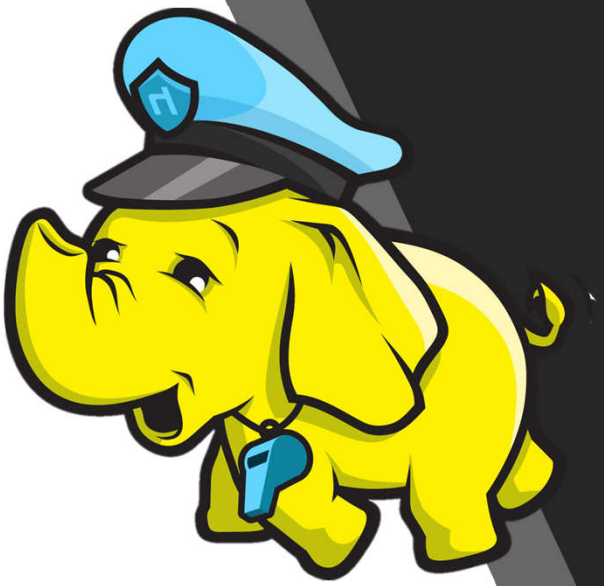
- Hive developed by Facebook is a data warehouse:
  - Built on top of Hadoop
  - Provides a simple language known as HiveQL similar to SQL for querying, summarization, and analysis.
  - Hive makes querying faster through indexing.

Apache Hadoop Ecosystem: Hive - MapReduce - HDFS



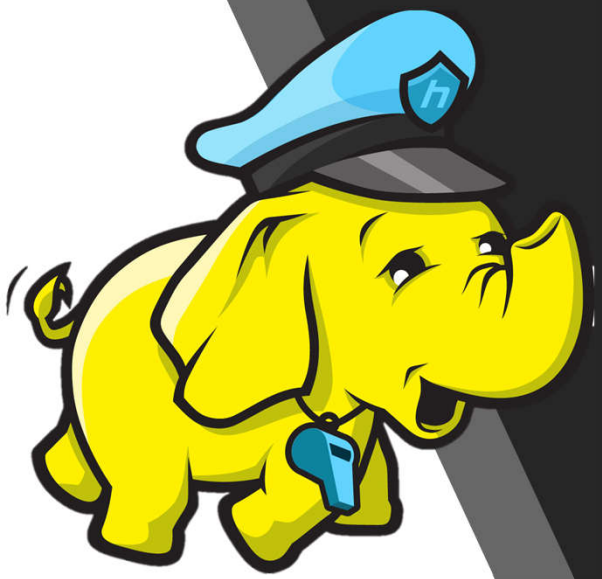
Stores schema information and provides a structure to data stored

Apache Hadoop Ecosystem: Spark



- Spark:
  - Like MapReduce: Distribute data across a cluster, and process that data in parallel.
  - Unlike MapReduce which shuffles files around on disk: Spark runs in memory, making much faster at processing data.
- Spark can run on Apache Hadoop clusters, its platforms
- It can access a large variety of data sources:
  - Hadoop Distributed File System
  - (HDFS)
  - Apache
  - Cassandra, Apache HBase
  - Cloud-based storage.

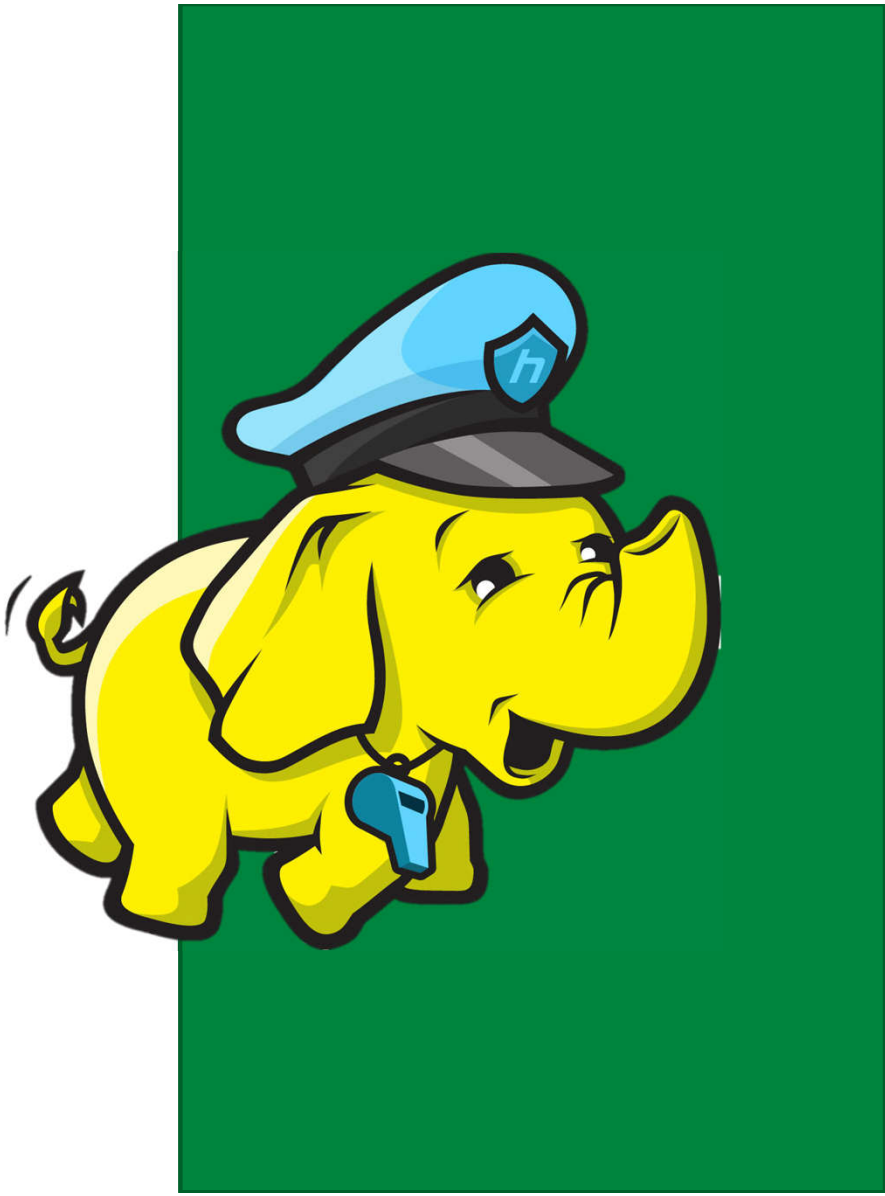
Apache Hadoop Ecosystem: Spark



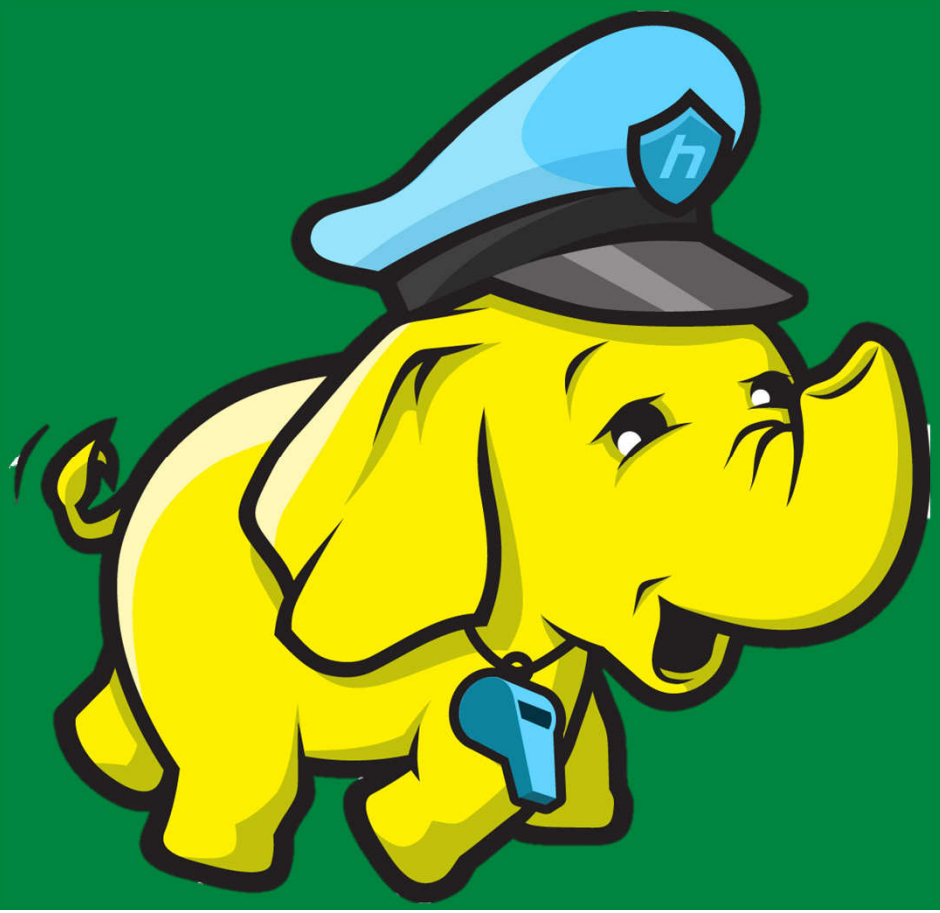
- Spark consists of a number of components:
  - Spark Core
  - Spark Streaming
  - Spark Machine Learning Library (MLlib)
  - Spark SQL + DataFrames
  - GraphX

Apache Hadoop Ecosystem: Why we use it

- Provides reliable shared storage (HDFS) and analysis system (MapReduce/Spark)
- Highly scalable
- Very cost-effective
- Highly flexible
- Has built-in fault tolerance.
- Works on the principle of write once and read multiple times.
- Optimized for large and very large data sets.



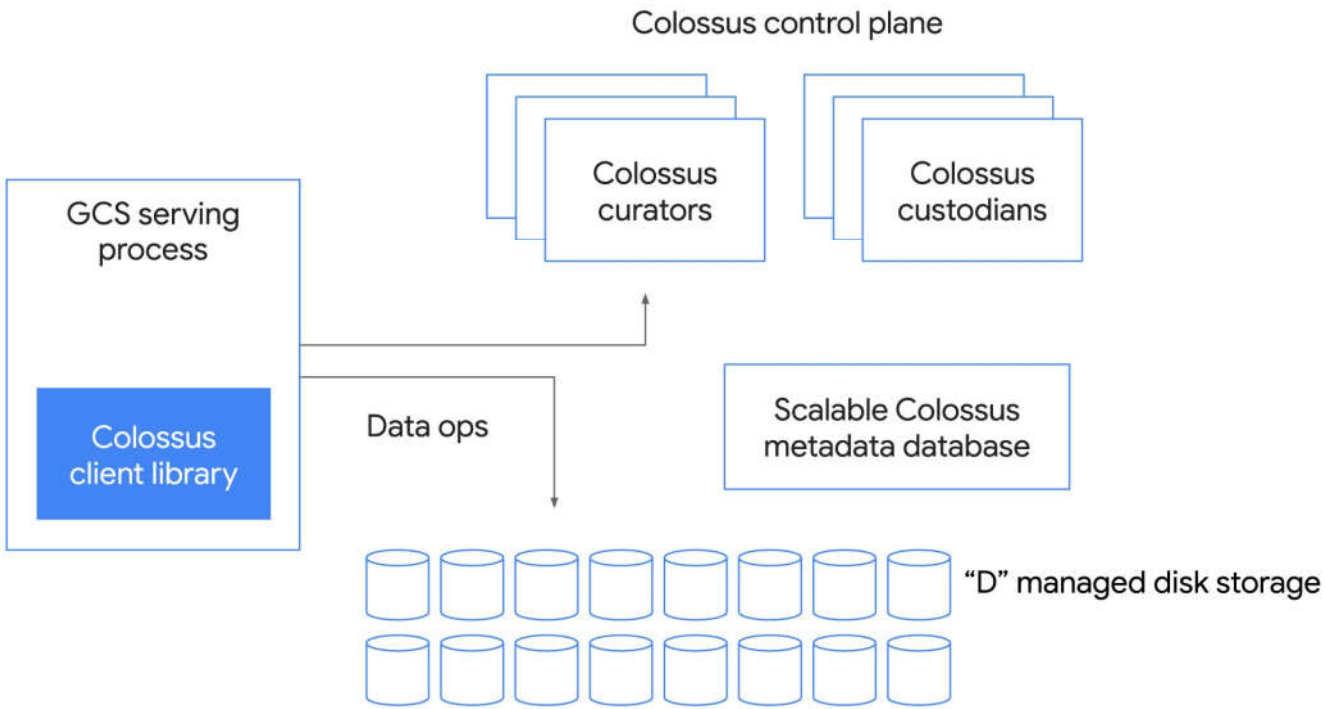
How does Hadoop fit into GCP



Understanding Colossus

- Next generation of the Google File System
- Enhances storage scalability
- Improves Availability
- Introduced a distributed metadata model

Understanding Colossus



Typical Cluster in Colossus