# Case Study: Hadoop

Introduction to Hadoop Environment
Data Flow
Hadoop I/O
Query languages for Hadoop
Hadoop and Amazon Cloud

# Hadoop

- Apache Hadoop is an open source distributed processing framework that manages data processing and storage for big data.
- Hadoop facilitates analyzing large amounts of data parallelly and more quickly.
- It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.
- Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.
- It makes use of MapReduce programming model to process data sets.

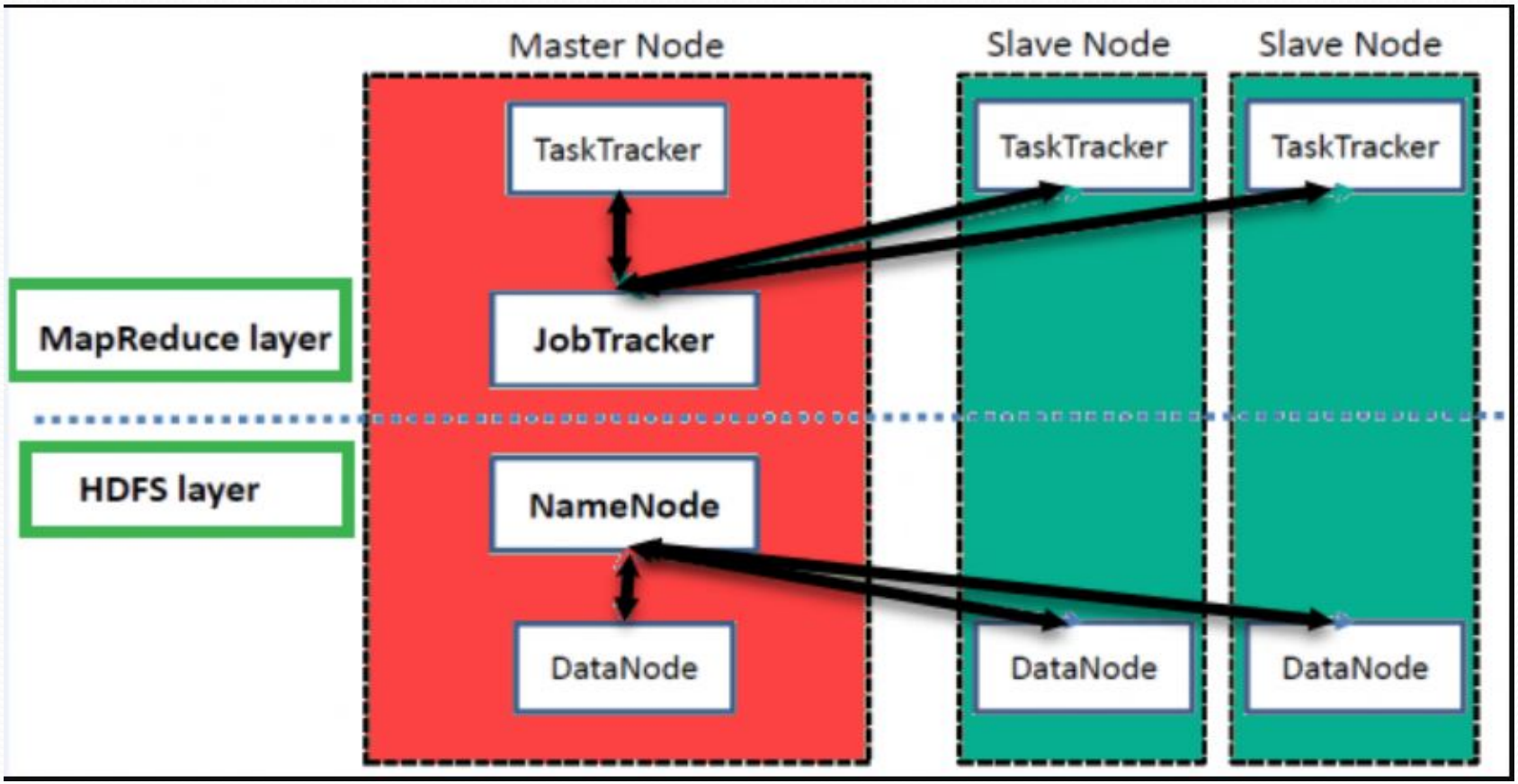
# Why to use Hadoop?

- It facilitates processing of huge data sets easily on large clusters of computers.
- It provides efficient, reliable and easy to use common infrastructure.
- It supports processing of data stored in each node, without necessity of moving data for initial processing.
- It does not rely up on hardware to provide fault tolerant and high availability.
- Servers can be added or removed dynamically without interrupting the operation of the system.
- It is open source, so is available freely.
- It is compatible on almost all the platforms.

# Hadoop components

- Hadoop consists of three main components.
  - HDFS: the storage unit in Hadoop
  - Map Reduce: the processing method
  - Yarn: the resource negotiator

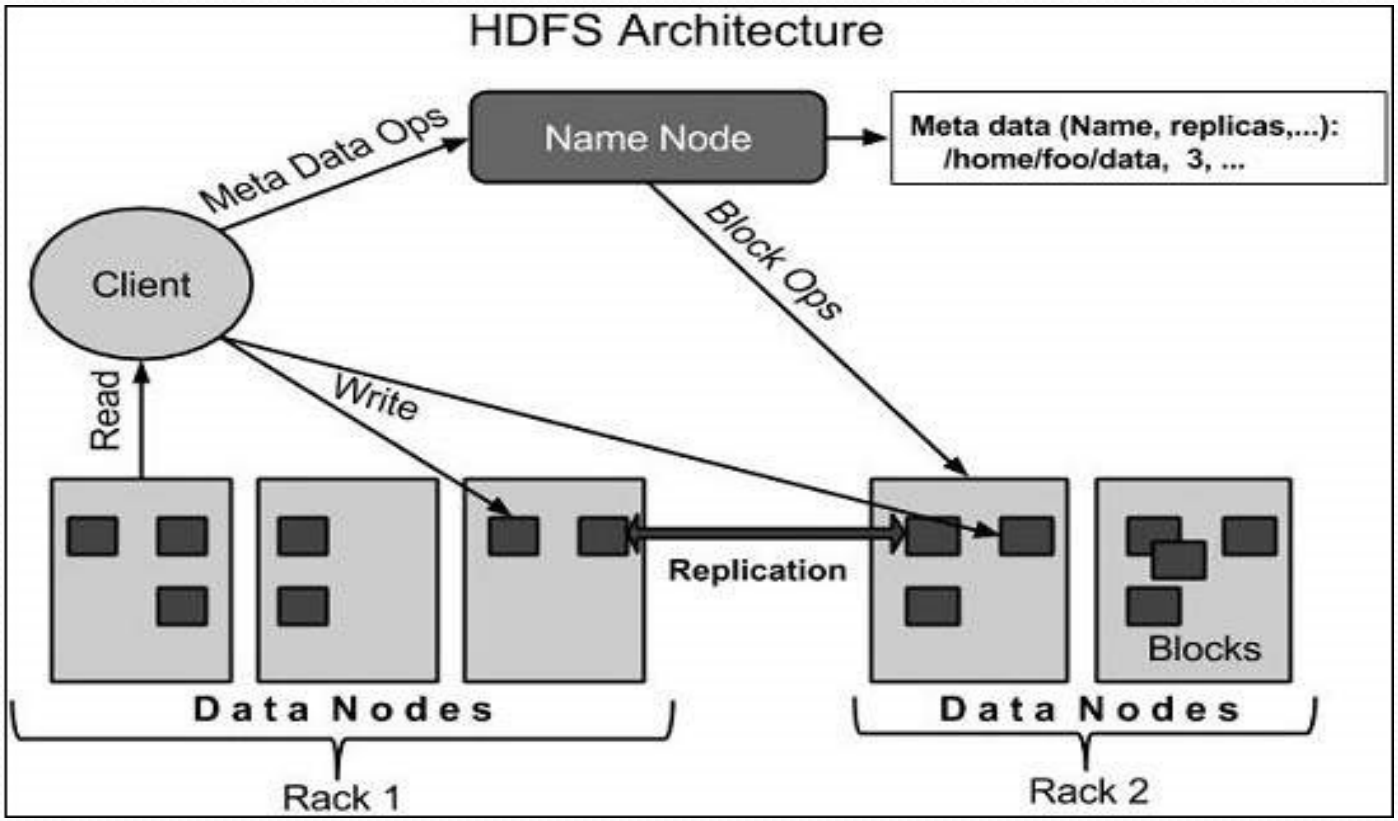
# Hadoop Architecture



# Hadoop Architecture

- Hadoop consists of mainly two main core components HDFS, MapReduce.
- HDFS is the **Hadoop Distributed File System** ( **HDFS** ) where the data is stored. It uses Master-Slave architecture to distribute, store and retrieve the data efficiently.
- MapReduce is a computational model and software framework for writing applications which are run on Hadoop. These MapReduce programs are capable of processing enormous data in parallel on large clusters of computation nodes.
- A Hadoop cluster consists of a single master and multiple slave nodes. The master node includes Job Tracker, Task Tracker, NameNode, and DataNode whereas the slave node includes DataNode and TaskTracker.

# HDFS Architecture



- **Hadoop Distributed File System**
  - The Hadoop Distributed File System (HDFS) is a distributed file system for Hadoop.
  - It contains a master/slave architecture. This architecture consist of a single NameNode performs the role of master, and multiple DataNodes performs the role of a slave.
  - Both NameNode and DataNode are capable enough to run on commodity machines.
  - The Java language is used to develop HDFS. So any machine that supports Java language can easily run the NameNode and DataNode software.

- **NameNode**
  - It is a single master server exist in the HDFS cluster.
  - As it is a single node, it may become the reason of single point failure.
  - It manages the file system namespace by executing an operation like the opening, renaming and closing the files.
  - It simplifies the architecture of the system.
- **DataNode**
  - The HDFS cluster contains multiple DataNodes.
  - Each DataNode contains multiple data blocks.
  - These data blocks are used to store data.
  - It is the responsibility of DataNode to read and write requests from the file system's clients.
  - It performs block creation, deletion, and replication upon instruction from the NameNode.

- **Job Tracker**
  - The role of Job Tracker is to accept the MapReduce jobs from client and process the data by using NameNode.
  - In response, NameNode provides metadata to Job Tracker.
- **Task Tracker**
  - It works as a slave node for Job Tracker.
  - It receives task and code from Job Tracker and applies that code on the file. This process can also be called as a Mapper.

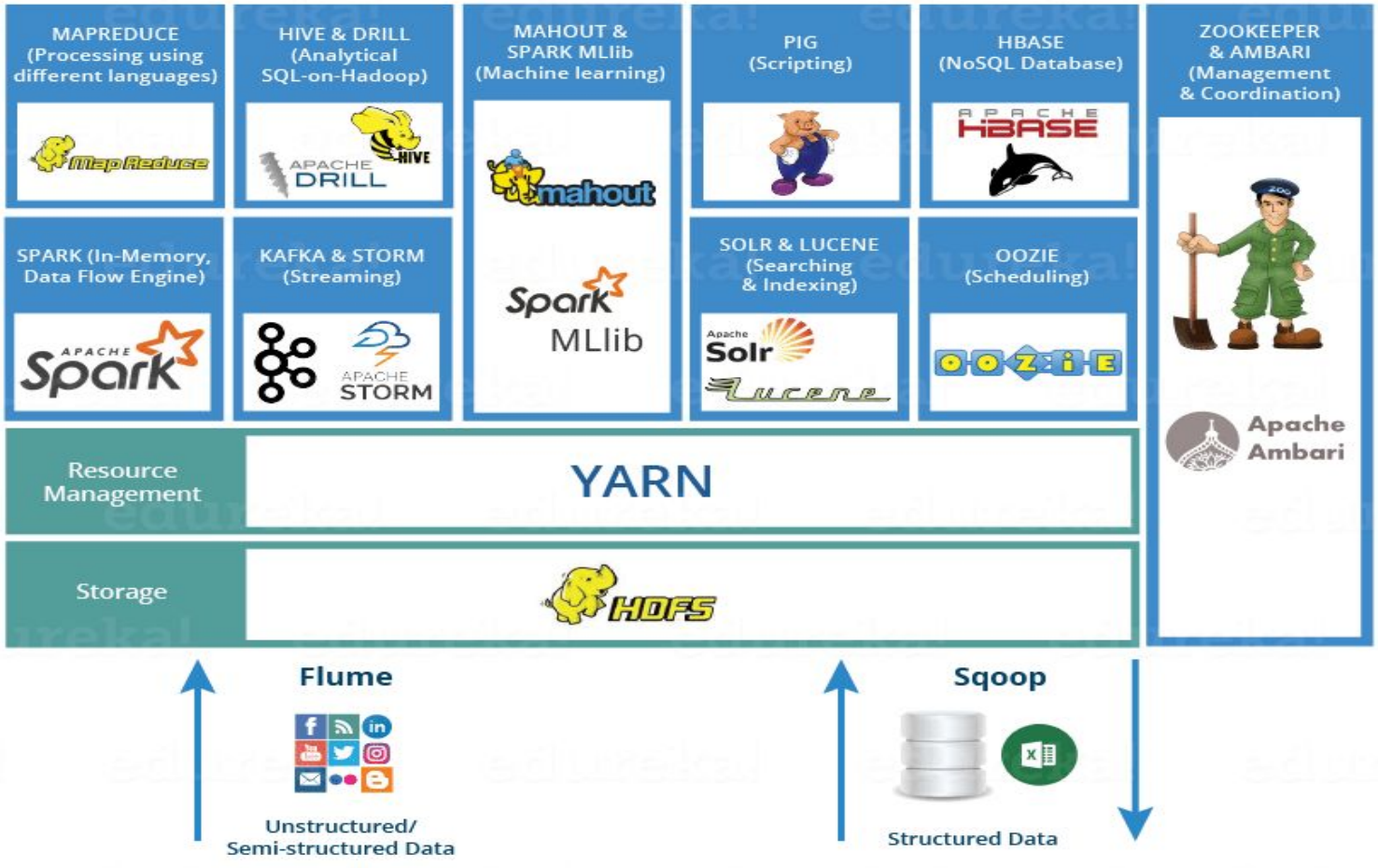
- **MapReduce Layer**
  - The MapReduce comes into existence when the client application submits the MapReduce job to Job Tracker. In response, the Job Tracker sends the request to the appropriate Task Trackers. Sometimes, the TaskTracker fails or time out. In such a case, that part of the job is rescheduled.

# How Hadoop works

- Hadoop stores and processes the data in a distributed manner across the cluster of commodity hardware.
- To store and process any data, the client submits the data and program to the Hadoop cluster.
- Hadoop HDFS stores the data, MapReduce processes the data stored in HDFS, and YARN divides the tasks and assigns resources.

- The Hadoop ecosystem consists of following components:
  - Hadoop Common
  - Hadoop Distributed File System (HDFS)
  - YARN
  - Hadoop MapReduce
  - Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.
  - Sqoop is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

# Hadoop Ecosystem



# Hadoop Common

- It consists of the common utilities that support other Hadoop modules.
- It is also known as Hadoop core.
- It provides file system and operating system level abstractions.
- It contains Java archive files and scripts needed to start Hadoop

# Hadoop Distributed File System (HDFS)

- It is a file system written in JAVA based on Google's GFS.
- It is a distributed, scalable and portable file system used by Hadoop.
- It is responsible to store data in the cluster.
- It provides redundant storage for huge data sets.
- The data files are split into blocks and are distributed across the nodes in the cluster.
- The details of which blocks forms a file and where they are stored is tracked by the name node.
- The default replication of data in HDFS is 3.
- When a client wants to retrieve data, it communicates with name node to determine the blocks and node storing them; and then the client directly communicates with the data node to read the data.

# Yet Another Resource Negotiator (YARN)

- It provides the solution to the scalability problems of classical MapReduce.
- It is a framework for job scheduling and cluster resource management.
- It enables Hadoop to support more varied processing approach and a broader array of applications

# Query Language for Hadoop

- **Hive**
  - Hive is the query language used by Hadoop.
  - It is used to query and manage large data sets of a Hadoop cluster using an SQL like language called HiveQL.

# ZooKeeper

- ZooKeeper is a distributed coordination service that is often used in conjunction with the Hadoop ecosystem. It is used to manage distributed systems, such as maintaining configuration information, providing distributed synchronization, and providing group services.
- In a Hadoop ecosystem, ZooKeeper is often used to coordinate and manage the different components of the system, such as the NameNode, DataNode, and JobTracker. For example, it can be used to maintain the state of the HDFS cluster and ensure that only one NameNode is active at a time. It can also be used to manage the allocation of resources in a YARN cluster.
- Additionally, ZooKeeper can be used as a centralized configuration management tool for Hadoop services, and it can be used for leader election for distributed systems, this is particularly useful for services like Storm, Kafka, and HBase.
- ZooKeeper is a distributed coordination service that provides a centralized and consistent way to manage and coordinate distributed systems in the Hadoop ecosystem. It helps to maintain the state of the cluster, allocate resources, and provide other services that are needed to run a Hadoop cluster smoothly.

# GFS VS. HDFS

| GFS | HDFS |
|---|---|
| Master | NameNode |
| ChunkServer | DataNode |
| Operation Log | Journal, Edit Log |
| Chunk | Block |
| Random file writes possible | Only append is possible |
| Multiple writer/reader model | Single writer/multiple reader model |
| Default chunk size: 64MB | Default block size: 128MB |

17

# Hadoop : Data Replication and Consistency

- Hadoop is an open-source distributed computing system that is designed to handle big data workloads. It uses a distributed file system called HDFS (Hadoop Distributed File System) to store and manage large amounts of data.
- Hadoop handles data replication by creating multiple copies of data blocks and storing them on different servers in the cluster. By default, HDFS replicates each data block three times across the cluster, but this can be configured to be more or less depending on the needs of the application.

# Hadoop : Data Replication and Consistency

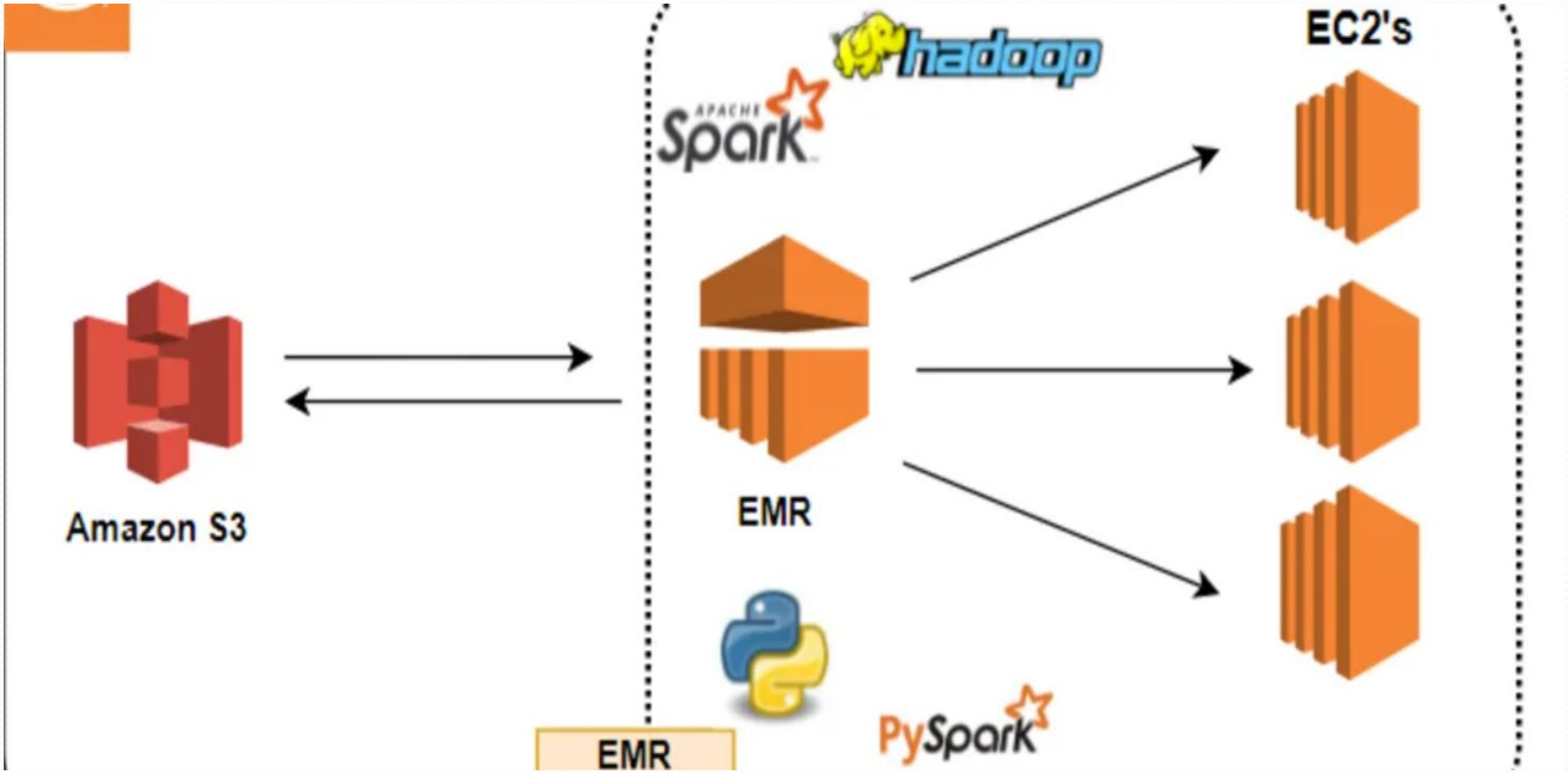
- Hadoop ensures data consistency by using a technique called "write-once-read-many" (WORM). This means that once a data block is written to HDFS, it cannot be modified or deleted. If a client wants to update a file, it must write a new version of the file, and the old version remains available for reading.
- Additionally, Hadoop uses a technique called "block checksum" to ensure data integrity. Each data block is assigned a unique checksum at the time of writing, and this checksum is verified before the block is read. If the checksum does not match the expected value, the data block is considered corrupt and is replaced with a replica from another node.
- Hadoop handles data replication by creating multiple copies of data blocks and storing them on different servers in the cluster. It ensures data consistency by using a "write-once-read-many" model, and by using block checksum to ensure data integrity.

# Hadoop and Amazon Cloud

- Amazon cloud provides an interface to create and manage fully configured, elastic clusters of instances running Hadoop and other applications in the Hadoop ecosystem.
- Amazon EMR securely and reliably handles a broad set of big data use cases, including log analysis, web indexing, data transformations (ETL), machine learning, financial analysis, scientific simulation, and bioinformatics.
- It uses Amazon EMR to easily install and configure tools such as Hive, Pig, HBase and many other tools on the personal cluster.

# Hadoop and Amazon Cloud

- In the Hadoop project, Amazon EMR programmatically installs and configures applications including Hadoop MapReduce, YARN and HDFS, across the nodes in the cluster.
- By using EMR file system on Amazon cluster, one can leverage Amazon S3 as the data layer for Hadoop.
- By storing data on Amazon S3, one can decouple compute layer from storage layer, allowing one to size the Amazon EMR cluster for the amount of CPU and memory required for the workloads instead of having extra nodes in the cluster to maximize on-cluster storage.
- EMR file system is customized for Hadoop to directly read and write in parallel to Amazon S3 performance.
- Hadoop in Amazon EMR can be used as an elastic query layer.


EC2's
hadoop
APACHE
Spark
Amazon S3
EMR
EMR
PySpark

# Advantages of Hadoop on Amazon EMR

- **Increased speed and agility**
  - One can initialize a new Hadoop cluster dynamically and quickly, or add servers to the existing Amazon cluster.
  - It reduces time taken for the resources to be available to the users.
  - Using Hadoop on Amazon AWS lowers cost and time it takes to allocate resources.
- **Reduced administrative complexity**
  - Amazon EMR automatically addresses the Hadoop infrastructure requirements, that reduce the complexity to administration.

# Advantages of Hadoop on Amazon EMR

- **Integration to other cloud services**
  - The Hadoop environment in Amazon cloud can be integrated with other cloud services like Amazon S3, Amazon DynamoDB and so on easily.
- **Flexible capacity**
  - With Amazon EMR, one can create clusters with the required capacity within minutes and use auto scaling to dynamically scale out and scale in nodes.

# References


- https://medium.com/geekculture/introduction-to-hadoop-architecture-83910caabc05
- https://medium.com/plumbersofdatascience/hdfs-architecture-in-depth-1edb822b95fa
- https://www.guru99.com/learn-hadoop-in-10-minutes.html
- https://www.geeksforgeeks.org/hadoop-architecture/
- https://medium.com/@zaman.nuces/apache-hive-a-brief-introduction-series-1-of-3-54ec07753dd2
- https://medium.com/nerd-for-tech/hive-commands-b5b6957d8820