**HADOOP**

**INTRODUCTION TO HADOOP** Hadoop is an open-source framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from a single server to thousands of machines, each offering local computation and storage. Rather than relying on hardware to deliver high availability, the framework itself is designed to detect and handle failures at the application layer, making it highly fault-tolerant.

Hadoop is composed of four main modules:

1. **Hadoop Common**: The common utilities that support the other Hadoop modules.
2. **Hadoop Distributed File System (HDFS)**: A distributed file system that provides high-throughput access to application data.
3. **Hadoop YARN**: A resource management platform responsible for managing compute resources in clusters and using them for scheduling users' applications.
4. **Hadoop MapReduce**: A programming model for large-scale data processing.

Hadoop is widely used in big data applications, such as data storage, data processing, and data analysis, making it a cornerstone in the big data ecosystem. Its ability to process vast amounts of data quickly and efficiently has led to its adoption in various industries, including finance, healthcare, retail, and technology.

**HISTORY OF HADOOP**

Hadoop, an open-source framework for distributed data processing, originated from the Nutch project, an open-source web search engine created by Doug Cutting and Mike Cafarella.

1. **Early Origins and Nutch (2002-2004**)

Hadoop began as part of the Nutch project, an open-source web search engine created by Doug Cutting and Mike Cafarella. As the internet grew, Nutch struggled with the increasing amount of data, prompting the need for a more scalable solution.

2. **Influence of Google (2003)**

Google's papers on the Google File System (GFS) and MapReduce heavily influenced the development of Hadoop. Cutting and Cafarella adapted these concepts to handle large-scale data processing within Nutch.

**3. Birth of Hadoop (2005-2006)**

In 2005, the key components of Nutch were separated to form Hadoop, named after Cutting’s son’s toy elephant. By 2006, Hadoop had become an official project under the Apache Software Foundation, marking its formal beginning.

**4. Growth and Adoption (2007-2010)**

Yahoo! played a crucial role in Hadoop’s early development, using it to power its web search infrastructure. This success led to wider adoption by major companies like Facebook, LinkedIn, and eBay, solidifying Hadoop’s place in big data processing.

**5. Formation of the Ecosystem (2010-present)**

As Hadoop gained popularity, a range of complementary tools such as HBase, Hive, Pig, and Spark were developed, expanding its capabilities. The project was eventually split into multiple sub-projects to better manage its growing ecosystem.

**VERSIONS OF HADOOP**

Hadoop has undergone several versions since its inception, each introducing new features and improvements.

**1. Hadoop 0.x Series**

* **Hadoop 0.1 (2006)**: The first official release of Hadoop after it became an Apache project.
* **Hadoop 0.20.x (2009)**: Significant release with improvements in the MapReduce framework and HDFS.

**2. Hadoop 1.x Series (2011-2012)**

* **Hadoop 1.0.0 (2011)**: Introduced stable MapReduce and HDFS features. Marked as the first major stable version.
* **Hadoop 1.2.x**: Continued improvements in stability, performance, and bug fixes.

**3. Hadoop 2.x Series (2013-2016)**

* **Hadoop 2.0.0 (2013)**: Introduction of YARN (Yet Another Resource Negotiator), which decoupled the resource management and scheduling capabilities from the MapReduce engine.
* **Hadoop 2.2.0 (2013)**: First stable release in the 2.x series, with full YARN integration.
* **Hadoop 2.6.x (2014-2015)**: Enhanced HDFS with features like support for heterogeneous storage (e.g., SSDs) and improvements in scalability.

**4. Hadoop 3.x Series (2017-Present)**

* **Hadoop 3.0.0 (2017)**: Major release introducing HDFS erasure coding (reducing storage overhead), support for more than 2 NameNodes, and native support for containerized workloads (Docker and other container technologies).
* **Hadoop 3.1.x (2018-2019)**: Added support for GPU scheduling and various optimizations in YARN and HDFS.
* **Hadoop 3.2.x (2019-2021)**: Continued enhancements, including improvements in cloud storage support, native support for Java 11, and better integration with other big data tools.

**5. Hadoop 3.3.x Series (2020-Present)**

* **Hadoop 3.3.x (2020-Present)**: Ongoing development with enhancements like better container management in YARN, improved compatibility with Kubernetes, and further optimizations in HDFS and YARN performance.

**Hardware Requirements**

**Minimum Requirements (For Development/Small Scale Testing)**

* **CPU**: Dual-core processor
* **RAM**: 8 GB
* **Storage**: 100 GB
* **Network**: Gigabit Ethernet

**Recommended Requirements (For Production/Cluster Deployment)**

* **Master Node (NameNode, ResourceManager)**
  + **CPU**: Quad-core processor or better
  + **RAM**: 16-64 GB (depends on cluster size)
  + **Storage**: SSDs recommended for faster access, 1 TB or more
  + **Network**: High-speed network (Gigabit Ethernet or higher)
* **Slave Nodes (DataNodes, NodeManagers)**
  + **CPU**: Quad-core processor or better
  + **RAM**: 16-64 GB (depends on workload)
  + **Storage**: Multiple high-capacity disks (HDDs or SSDs), 1 TB or more
  + **Network**: High-speed network (Gigabit Ethernet or higher)

**Software Requirements**

**Operating System**

* **Linux** (Preferred distributions: CentOS, Ubuntu, Debian, Red Hat)
* **Windows** (Supported but less common in production environments)

**Java**

* **Java Development Kit (JDK)**: Version 8 or higher (Hadoop is primarily developed in Java)
  + Make sure JAVA\_HOME environment variable is set correctly.

**SSH**

* **Secure Shell (SSH)**: Password-less SSH access must be configured between nodes in the cluster.

**Hadoop Distribution**

* **Hadoop Version**: Ensure compatibility with other software components.
  + Download the appropriate Hadoop version from [Apache Hadoop](https://hadoop.apache.org/).

**Configuration Recommendations**

* **Memory Allocation**: Allocate enough memory to HDFS and YARN based on the available RAM and expected workload.
* **Storage Configuration**: Use RAID configurations for data redundancy and performance improvements. SSDs are preferred for critical components like the NameNode.
* **Network Setup**: Ensure a high-speed, reliable network setup. In a large cluster, consider using multiple network interfaces for better performance.

**Example Deployment Sizes**

**Small Deployment**

* **Nodes**: 1-10
* **RAM per Node**: 16 GB
* **Storage per Node**: 1 TB
* **Use Case**: Development, testing, small-scale data processing

**Medium Deployment**

* **Nodes**: 10-50
* **RAM per Node**: 32 GB
* **Storage per Node**: 2-4 TB
* **Use Case**: Medium-scale data processing, business analytics

**Large Deployment**

* **Nodes**: 50-100+
* **RAM per Node**: 64 GB or more
* **Storage per Node**: 4-10 TB
* **Use Case**: Large-scale data processing, enterprise-level data analytics, real-time data processing