# <u>Big Data - Day 2</u>

**24th Sep, 2020**

## Cloud Computing

There are three main types of cloud computing infrastructure, depending on the client”s available infrastructure, IT personnel, cost, and security:

**Infrastructure as a Service (IaaS)**

With Infrastructure as a Service, the cloud provider owns and manages the hardware on which the software stack runs. The hardware includes servers, networking, and storage. On the other hand, the client manages the operating system, applications, databases, functions, as well as the security of the data.

Common examples are AWS, GCP, and Microsoft Azure.

**Platform as a Service (PaaS)**

Same as the IaaS, except that the service provider also provides and manages the operating system and the databases. This means that the client will be responsible for the applications, functions, and data.

Examples include AWS Elastic Beanstalk and Google App Engine.

**Software as a Service (SaaS)**

Software as a Service is one of the simplest for the end-user. The provider provides and manages everything. The client is only responsible for the data.

Popular examples include CRM software, cloud-based file-storage, and email.

Data

Data as a general concept is existing information or knowledge represented or coded in some form suitable for better usage or processing.

### Types of Data

There are three types of data:

1. *Structures data*

This is data that has been organized into a formatted repository that is typically a database in the form of tables of rows and columns. The tables have relational keys that map into predefined fields. An example is RDBMS using SQL.

2. *Semi-structured data*

   This is information that does not reside in a relational database but has some organizational properties that make it easy to analyze. With some processing, it can easily be stored in a relational database. Examples include JSON, CSV, XML, and TSV files.

3. *Unstructured data*

   This is data that is not organized in a predefined manner or does not have a predefined data model thus making it unfit for a relational database. It can be in the form of text, audio, or video files.

**OLTP vs OLAP**

When it comes to data processing there are two types, namely transactional (OLTP) and analytical (OLAP).

In general, OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.

*OLTP (On-Line Transactional Processing)*
- Characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE).
- Very fast query processing.
- Data integrity in a multi-access environment.
- Measured in transactions per second.
- RDBMS such as Oracle, MySQL, and Postgresql are OLTP systems
- NoSQL such as Cassandra and MongoDB are also OLTP systems.

*OLAP (On-Line Analytical Processing)*
- Characterized by a low volume of transactions.
- Queries are often complex and involve aggregations.
- Applications are widely used in Data Mining techniques.
- In OLAP databases there is aggregated, historical data, stored in multi-dimensional schemas.
- Measured in response time.
- Example: a bank soring years of customer historical records.

## Hadoop

Hadoop is a framework that allows for the distributed processing of large 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.

### HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It is similar to existing file systems such as NTFS or EXT4 but instead of having physical blocks of 4KB it has blocks of 128 MB which are suitable for large data sets.

### DAEMONS | Processes

Daemons are another name for processes and are written in Java. Apache Hadoop 2 consists of the following daemons:

- **NameNode**

  Works on the master system. Its primary purpose is to manage all metadata. Metadata is the list of files stored in the HDFS. It consists of the type of data, logs of transactions, time of access, and who accessed it. It is stored in memory.

- **DataNode**

  Works on the slave system. It takes instructions from the NameNode and serves the read/write request from the client. They usually possess high memory since the data is stored in the DataNode.

- **Secondary NameNode**

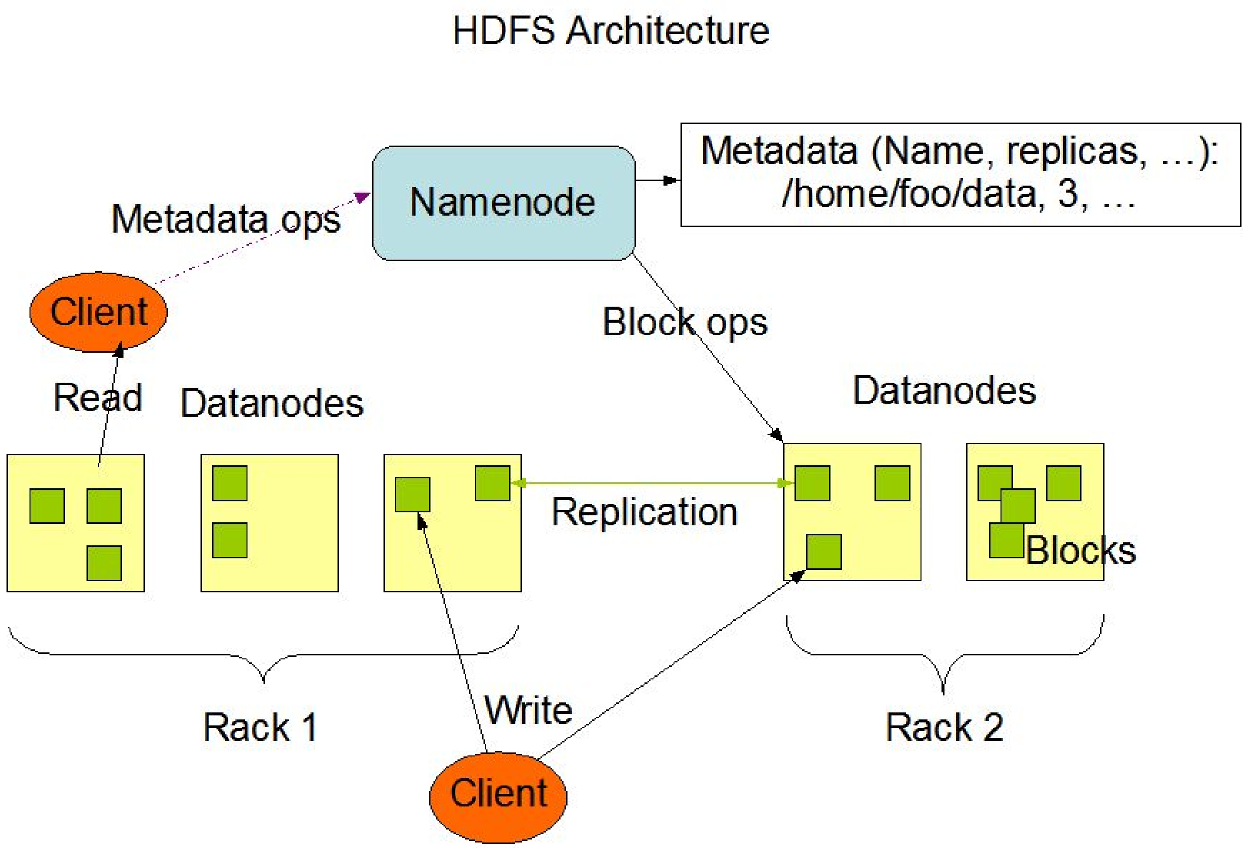
  Used for taking hourly backup or *checkpoint* of the metadata. In case of a Hadoop cluster failure, then this node takes the edit log file on disk and merges it with the *fsimage* file. This is the file to be transferred to a new system and a new Master is created with this metadata and the cluster made to run again properly.

- **Resource Manager**

  Also known as the Global Master Daemon works on the master system. It manages the resources for the applications that are running in a Hadoop cluster.
  It consists of 1) the Application master that is responsible for accepting requests from clients and allocating resources on the slave nodes to host the application master, and 2) the Scheduler for scheduling the resource availability.

- Node Manager

  Works on the Slave System and manages the memory resources within the Node and Memory Disk. Each Data Node has one Node Manager daemon running and is responsible for sending heartbeats to the Resource Manager. Other tasks include mapping, sorting, shuffling, and reducing.




- *Image courtesy of hadoop.apache.org*

**High Availability with Quorum Journal Manager and Zookeeper**

Before Hadoop 2.0.0, the NameNode was a single point of failure (SPOF) in the HDFS cluster. If that machine went down then the cluster as a whole became unavailable. This affected also times during maintenance.
Since Hadoop 2.0.0 two or more machines are configured as NameNodes with exactly one in an Active state and others on standby state. Both groups communicate with a group of separate daemons called JournalNodes or JNs which get durable copies of the log files from the Active

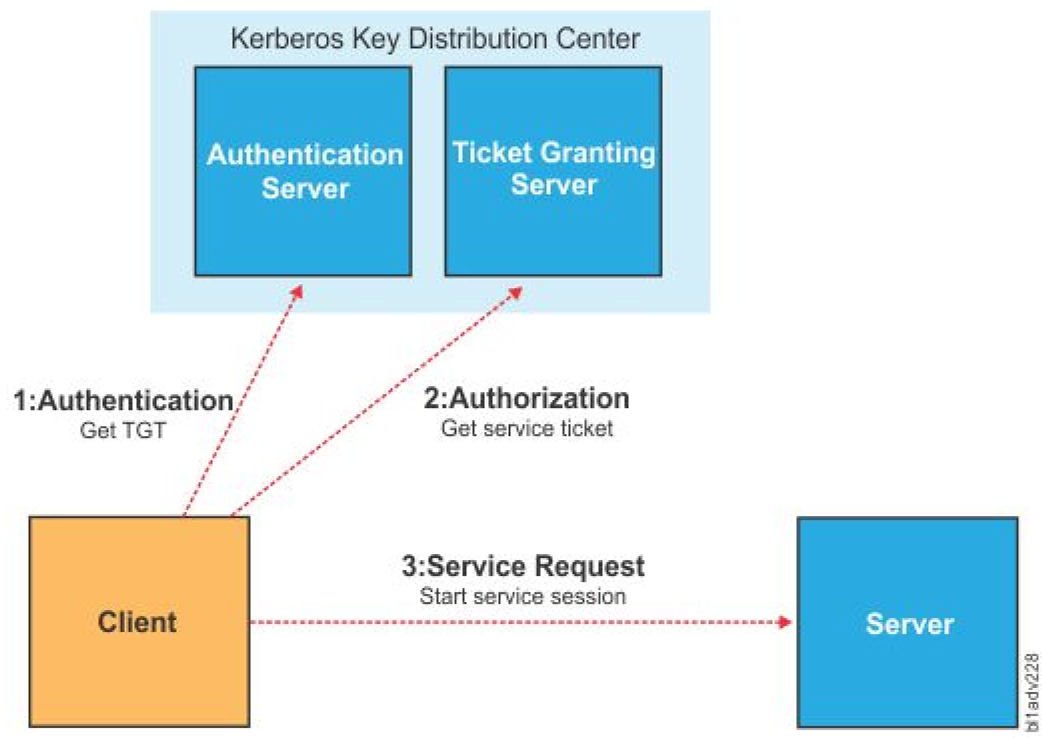
NameNode. The standby nodes also keep checking for changes by reading the log files and in the event of a failure can it has up-to-date information regarding the blocks in the clusters. The JournalNodes only allow a single NameNode to write at a time and during a failover, the NameNode with the most up-to-date information goes to an Active state.

Zookeeper is also a highly available service that maintains a small amount of coordination data, notifying clients of changes in the data, and monitoring clients for failures. Zookeeper does the following to make automatic failover possible:

- Failure detection - NameNodes maintain a persistent session in zookeeper. If a node fails the session expiries and other NameNodes are notified so that automatic failover is triggered.
- Active NodeName election - If an active NameNode crashes, another node may take a special exclusive lock in Zookeeper indicating that it should become the next active.
- 

**Kerberos**

Hadoop uses Kerberos for strong authentication and identity propagation for both users and services. Kerberos is a third party authentication mechanism, where users and services rely on the Kerberos server to authenticate one to the other.




The Kerberos server is also known as Key Distribution Center (KDC) and has three parts:

1. A database of users and services (known as principles) and their respective passwords

2. Authentication Server (AS) - performs the initial authentication and issues a Ticket Granting Ticket (TGT)
3. Ticket Granting Server (TGS) - issues subsequent tickets based on the initial TGT.

The user principle requests authentication from the AS which returns a TGT which is decrypted using the principles password known only to the user principle and the server. The user principle uses the Kerberos password to decrypt the TGT, and from then on can obtain service tickets from the TGS until the ticket expires.

## Hadoop Ecosystem cont

### Sqoop

- A tool designed to transfer data between Hadoop and a relational database
- Used to import data from relational databases such as Oracle or MySQL to HDFS and export data from HDFS to the relational databases

### HIVE

- Executes queries using MapReduce using SQL commands and scripts.
- Suitable for structured data.

### Flume

- A distributed service that collects event data such as streaming data, sensor data, or log files and transfers it to HDFS.
- It can collect data from multiple streams.

### Pig

- Pig converts its SQL-like scripts called Pig-Latin to Map and Reduce code, thereby saving users from writing complex MapReduce programs in Java.
- It is highly extensible with user-defined functions (UDFs).
- Ideal for interactive analysis with low latency which can be measured in milliseconds.

### HBase

- A NoSQL database or non-relational database.
- It provides support for a high volume of data and a high throughput.
- In an HBase, a table can have thousands of columns (like BigTable by Google)
- Can do CRUD (Create, Read, Update, and Delete)