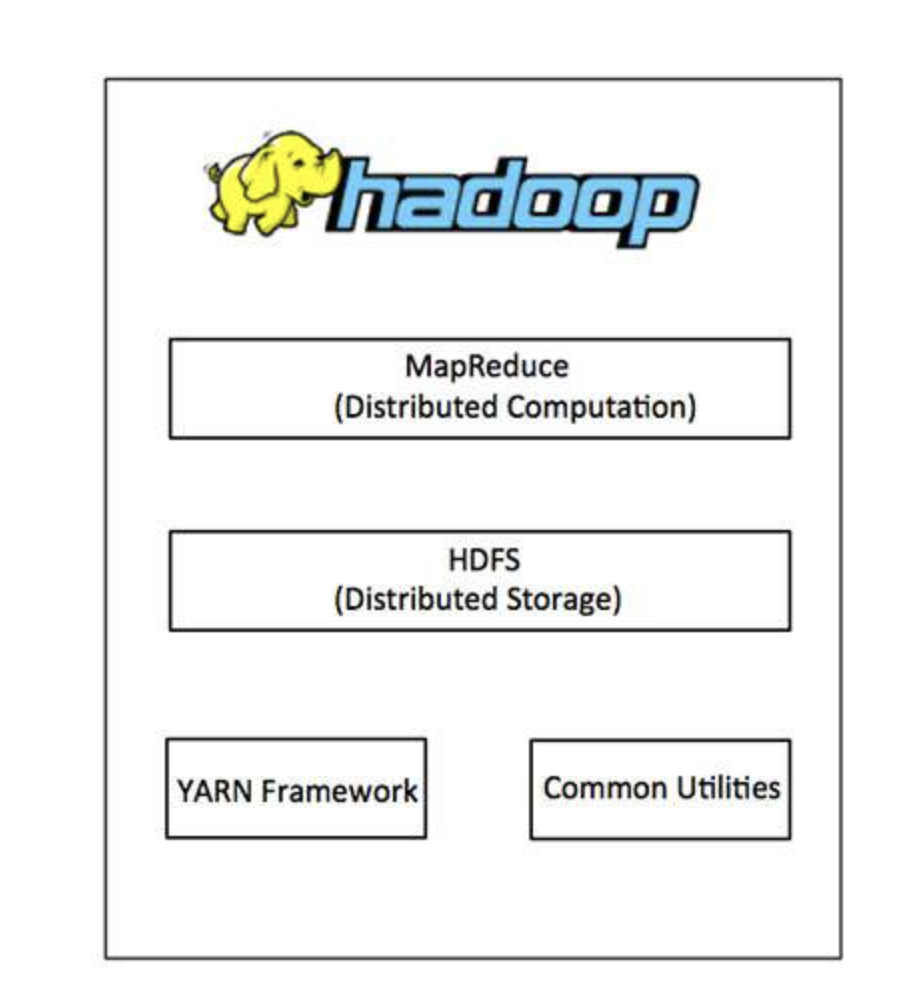
**UNIT – 3( HADOOP)**

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed *storage* and *computation* across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Hadoop Architecture

At its core, Hadoop has two major layers namely −

* Processing/Computation layer (MapReduce), and
* Storage layer (Hadoop Distributed File System).

****

## Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. It is highly fault-tolerant and is designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications having large datasets.

Apart from the above-mentioned two core components, Hadoop framework also includes the following two modules −

* **Hadoop Common** − These are Java libraries and utilities required by other Hadoop modules.
* **Hadoop YARN** − This is a framework for job scheduling and cluster resource management.

Hadoop is a framework that uses distributed storage and parallel processing to store and manage big data. It is the software most used by data analysts to handle big data, and its[market size continues to grow.](https://www.businesswire.com/news/home/20201112005841/en/Global-Hadoop-Market-2020-to-2027---by-Component-Deployment-Model-Organization-Size-and-End-user---ResearchAndMarkets.com) There are three components of Hadoop:

1. Hadoop HDFS - [Hadoop Distributed File System (HDFS)](https://www.simplilearn.com/tutorials/hadoop-tutorial/hdfs) is the storage unit.
2. Hadoop MapReduce - [Hadoop MapReduce](https://www.simplilearn.com/tutorials/hadoop-tutorial/mapreduce) is the processing unit.
3. Hadoop YARN - [Yet Another Resource Negotiator (YARN)](https://www.simplilearn.com/tutorials/hadoop-tutorial/yarn) is a resource management unit.

## How Does Hadoop Work?

It is quite expensive to build bigger servers with heavy configurations that handle large scale processing, but as an alternative, you can tie together many commodity computers with single-CPU, as a single functional distributed system and practically, the clustered machines can read the dataset in parallel and provide a much higher throughput. Moreover, it is cheaper than one high-end server. So this is the first motivational factor behind using Hadoop that it runs across clustered and low-cost machines.

Hadoop runs code across a cluster of computers. This process includes the following core tasks that Hadoop performs −

* Data is initially divided into directories and files. Files are divided into uniform sized blocks of 128M and 64M (preferably 128M).
* These files are then distributed across various cluster nodes for further processing.
* HDFS, being on top of the local file system, supervises the processing.
* Blocks are replicated for handling hardware failure.
* Checking that the code was executed successfully.
* Performing the sort that takes place between the map and reduce stages.
* Sending the sorted data to a certain computer.
* Writing the debugging logs for each job.

## Advantages of Hadoop

* Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
* Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.
* Servers can be added or removed from the cluster dynamically and Hadoop continues to operate without interruption.
* Another big advantage of Hadoop is that apart from being open source, it is compatible on all the platforms since it is Java based.

Hadoop File System was developed using distributed file system design. It is run on commodity hardware. Unlike other distributed systems, HDFS is highly fault tolerant and designed using low-cost hardware.

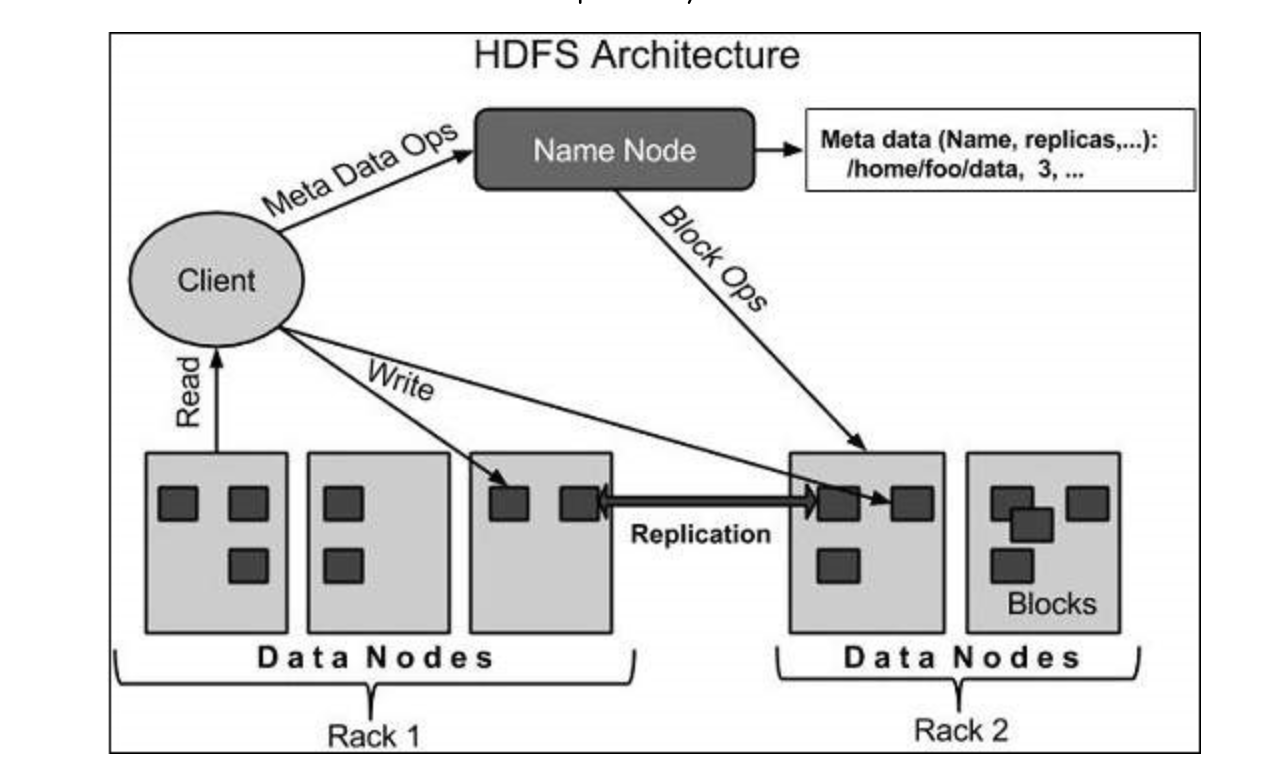
HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

Features of HDFS

* It is suitable for the distributed storage and processing.
* Hadoop provides a command interface to interact with HDFS.
* The built-in servers of namenode and datanode help users to easily check the status of cluster.
* Streaming access to file system data.
* HDFS provides file permissions and authentication.

HDFS Architecture

Given below is the architecture of a Hadoop File System.

****

HDFS follows the master-slave architecture and it has the following elements.

#### **Master and Slave Nodes**

Master and slave nodes form the [HDFS cluster](https://www.simplilearn.com/what-is-a-hadoop-cluster-article). The name node is called the master, and the data nodes are called the slaves.

### **Namenode**

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is a software that can be run on commodity hardware. The system having the namenode acts as the master server and it does the following tasks −

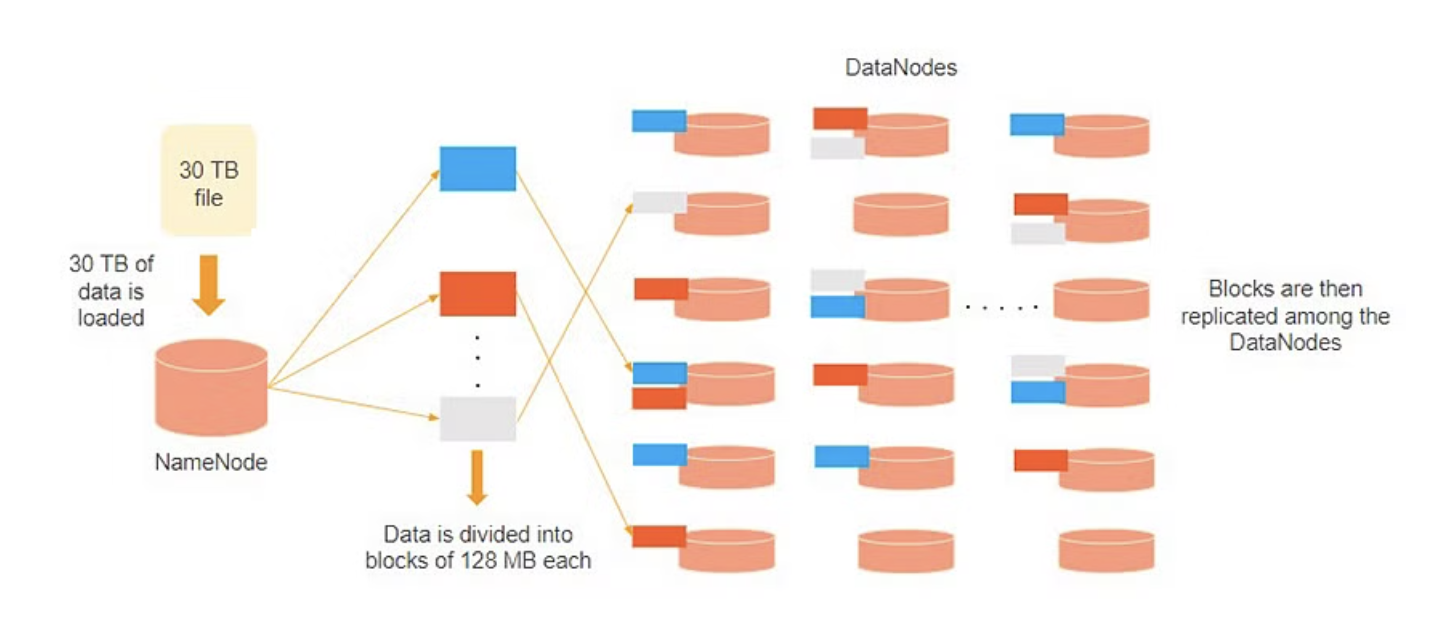
* Manages the file system namespace.
* Regulates client’s access to files.
* It also executes file system operations such as renaming, closing, and The name node is responsible for the workings of the data nodes. It also stores the metadata.

### **Datanode**

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system.

* Datanodes perform read-write operations on the file systems, as per client request.
* They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

The data nodes read, write, process, and replicate the data. They also send signals, known as heartbeats, to the name node. These heartbeats show the status of the data node.



Consider that 30TB of data is loaded into the name node. The name node distributes it across the data nodes, and this data is replicated among the data notes. You can see in the image above that the blue, grey, and red data are replicated among the three data nodes.

Replication of the data is performed three times by default. It is done this way, so if a commodity machine fails, you can replace it with a new machine that has the same data.

### **Block**

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

## Goals of HDFS

**Fault detection and recovery** − Since HDFS includes a large number of commodity hardware, failure of components is frequent. Therefore HDFS should have mechanisms for quick and automatic fault detection and recovery.

**Huge datasets** − HDFS should have hundreds of nodes per cluster to manage the applications having huge datasets.

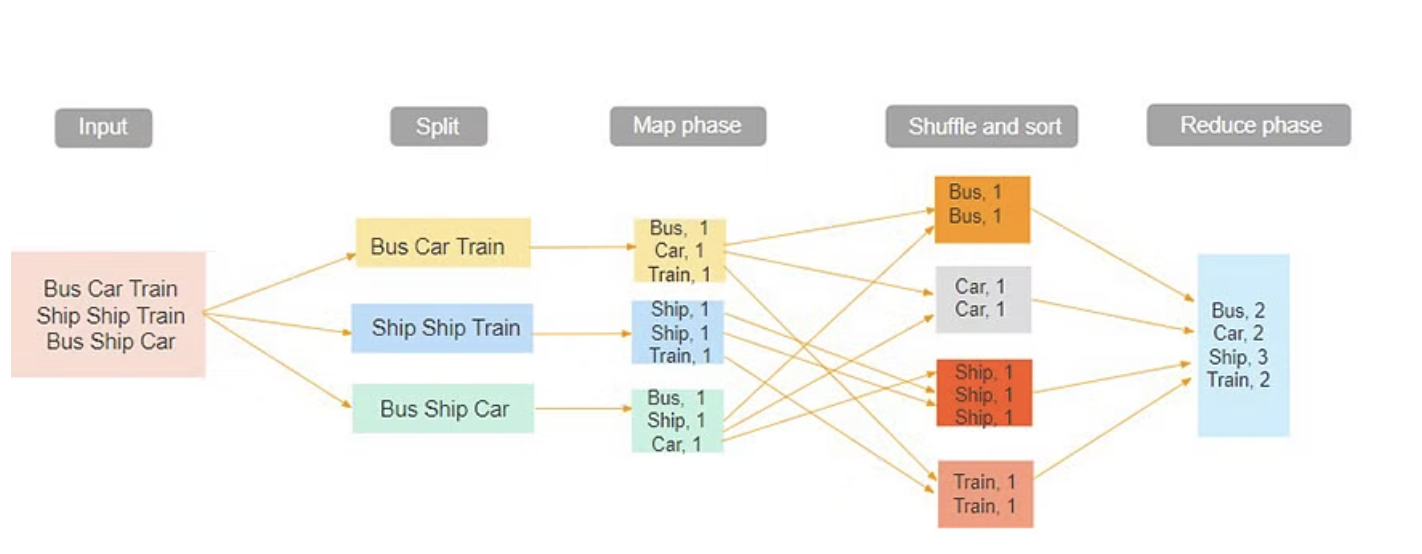
**Hardware at data** − A requested task can be done efficiently, when the computation takes place near the data. Especially where huge datasets are involved, it reduces the network traffic and increases the throughput.

## MapReduce

MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The MapReduce program runs on Hadoop which is an Apache open-source framework.

Hadoop MapReduce is the processing unit of Hadoop. In the MapReduce approach, the processing is done at the slave nodes, and the final result is sent to the master node.

A data containing code is used to process the entire data. This coded data is usually very small in comparison to the data itself. You only need to send a few kilobytes worth of code to perform a heavy-duty process on computers.



the input dataset is first split into chunks of data. In this example, the input has three lines of text with three separate entities - “bus car train,” “ship ship train,” “bus ship car.” The dataset is then split into three chunks, based on these entities, and processed parallelly.

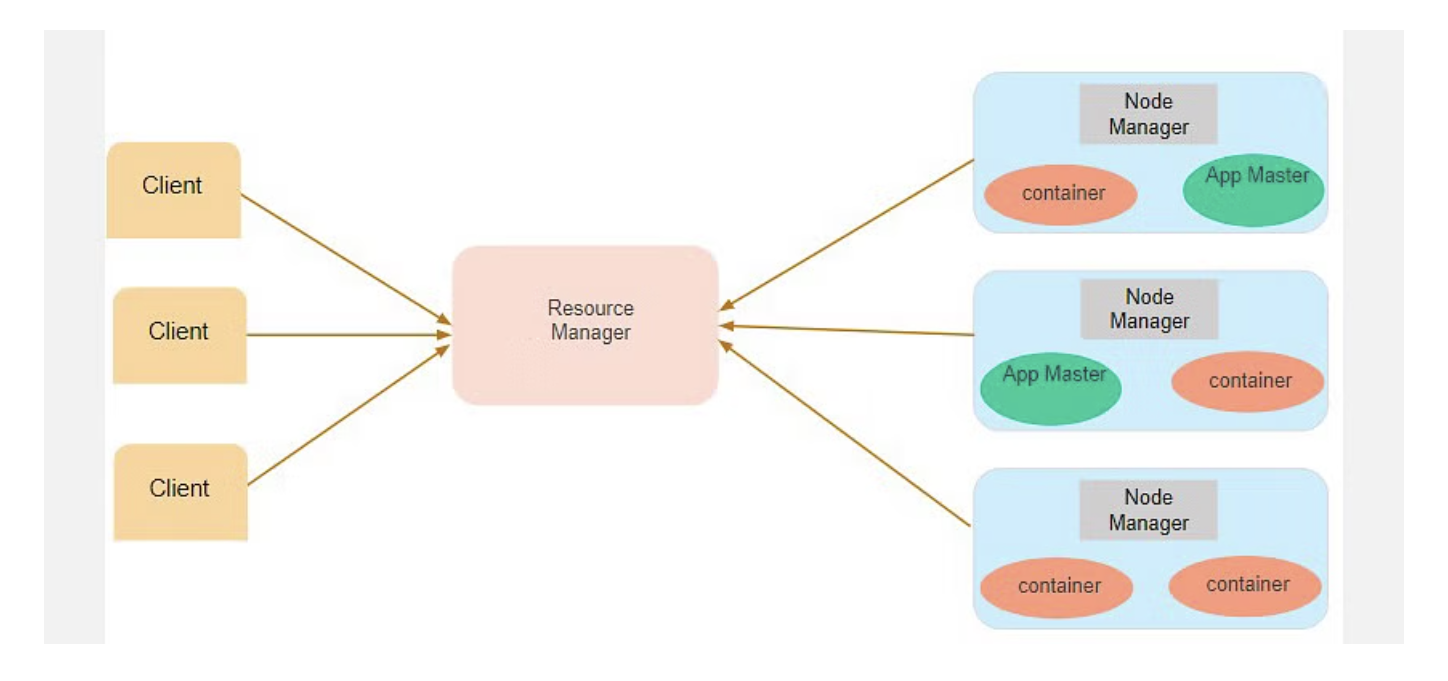
In the map phase, the data is assigned a key and a value of 1. In this case, we have one bus, one car, one ship, and one train.

These key-value pairs are then shuffled and sorted together based on their keys. At the reduce phase, the aggregation takes place, and the final output is obtained.

### **Hadoop YARN**

Hadoop YARN stands for Yet Another Resource Negotiator. It is the resource management unit of Hadoop and is available as a component of Hadoop version 2.

* Hadoop YARN acts like an OS to Hadoop. It is a file system that is built on top of HDFS.
* It is responsible for managing cluster resources to make sure you don't overload one machine.
* It performs job scheduling to make sure that the jobs are scheduled in the right place



Suppose a client machine wants to do a query or fetch some code for [data analysis](https://www.simplilearn.com/data-analysis-methods-process-types-article). This job request goes to the resource manager (Hadoop Yarn), which is responsible for resource allocation and management.

In the node section, each of the nodes has its node managers. These node managers manage the nodes and monitor the resource usage in the node. The containers contain a collection of physical resources, which could be RAM, CPU, or hard drives. Whenever a job request comes in, the app master requests the container from the node manager. Once the node manager gets the resource, it goes back to the Resource Manager.

# **Hadoop - HDFS Operations**

## Starting HDFS

Initially you have to format the configured HDFS file system, open namenode (HDFS server), and execute the following command.

$ hadoop namenode -format

After formatting the HDFS, start the distributed file system. The following command will start the namenode as well as the data nodes as cluster.

$ start-dfs.sh

## Listing Files in HDFS

After loading the information in the server, we can find the list of files in a directory, status of a file, using **‘ls’**. Given below is the syntax of **ls** that you can pass to a directory or a filename as an argument.

$ $HADOOP\_HOME/bin/hadoop fs -ls <args>

## Inserting Data into HDFS

Assume we have data in the file called file.txt in the local system which is ought to be saved in the hdfs file system. Follow the steps given below to insert the required file in the Hadoop file system.

### **Step 1**

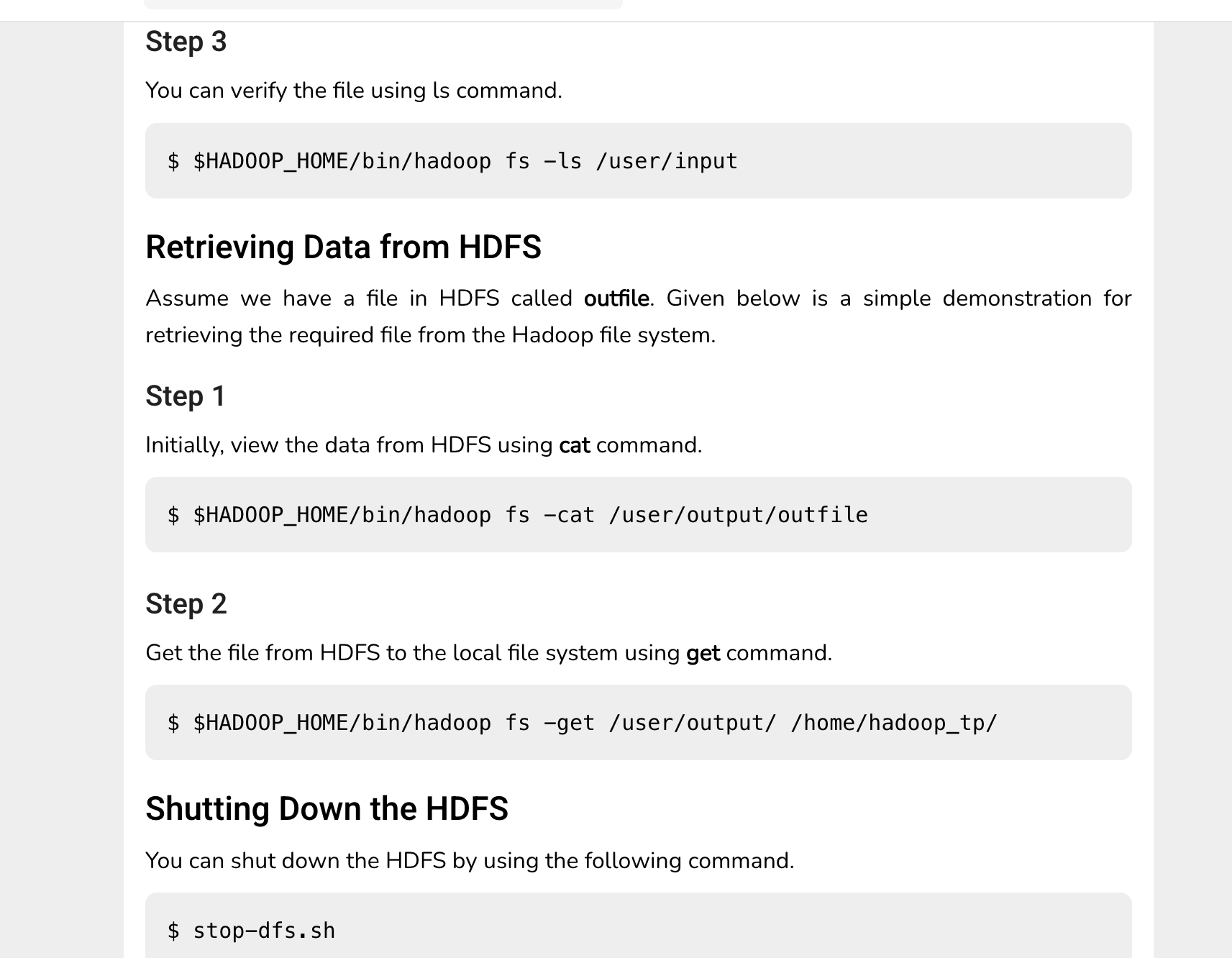
You have to create an input directory.

$ $HADOOP\_HOME/bin/hadoop fs -mkdir /user/input

### **Step 2**

Transfer and store a data file from local systems to the Hadoop file system using the put command.

$ $HADOOP\_HOME/bin/hadoop fs -put /home/file.txt /user/input

****

# **Hadoop - Environment Setup**

Hadoop is supported by GNU/Linux platform and its flavours. Therefore, we have to install a Linux operating system for setting up Hadoop environment. In case you have an OS other than Linux, you can install a VirtualBox software in it and have Linux inside the VirtualBox.

## Pre-installation Setup

Before installing Hadoop into the Linux environment, we need to set up Linux using **ssh** (Secure Shell). Follow the steps given below for setting up the Linux environment.

### **Creating a User**

At the beginning, it is recommended to create a separate user for Hadoop to isolate Hadoop file system from Unix file system. Follow the steps given below to create a user −

* Open the root using the command “su”.
* Create a user from the root account using the command “useradd username”.
* Now you can open an existing user account using the command “su username”.

Open the Linux terminal and type the following commands to create a user.

$ su

password:

# useradd hadoop

# passwd hadoop

New passwd:

Retype new passwd

## SSH Setup and Key Generation

SSH setup is required to do different operations on a cluster such as starting, stopping, distributed daemon shell operations. To authenticate different users of Hadoop, it is required to provide public/private key pair for a Hadoop user and share it with different users.

The following commands are used for generating a key value pair using SSH. Copy the public keys form id\_rsa.pub to authorized\_keys, and provide the owner with read and write permissions to authorized\_keys file respectively.

$ ssh-keygen -t rsa

$ cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

$ chmod 0600 ~/.ssh/authorized\_keys

**….. Complete the Hadoop installation setup**