Experiment no: 01

Title: Hadoop HDFS Practical:
-HDFS Basics, Hadoop Ecosystem Tools Overview.
-Installing Hadoop.
-Copying File to Hadoop.
-Copy from Hadoop File system and delete file.
-Moving and displaying files in HDFS.
-Programming exercises on Hadoop

-HDFS Basics, Hadoop Ecosystem Tools Overview.
HDFS Basics

HDFS (Hadoop Distributed File System) is the primary storage system used by Hadoop applications. It is designed to handle large datasets, providing high throughput access to data and being highly fault-tolerant. Below are some key features and components of HDFS:

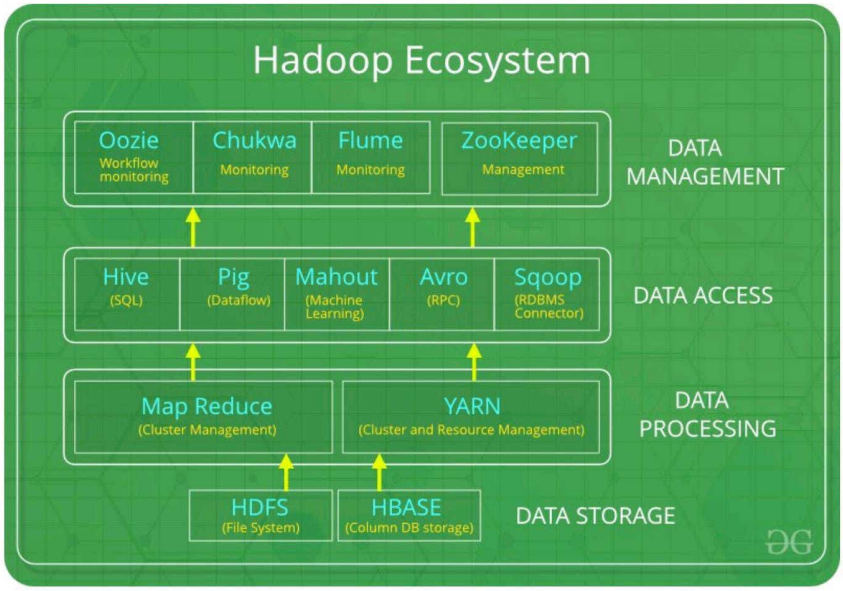
1. **Distributed Storage**: HDFS stores data across multiple machines in a cluster, ensuring that the data is both scalable and resilient to failures.
2. **Blocks**: HDFS splits large files into blocks (default size is 128MB or 256MB) and distributes them across the cluster. Each block is replicated (default replication factor is 3) to handle hardware failures.
3. **NameNode and DataNodes**:
    - **NameNode**: Manages the metadata of the files (like the directory structure and file names). It keeps track of which blocks make up a file and where those blocks are located across the cluster.
    - **DataNodes**: Store the actual data blocks. They regularly report back to the NameNode with the status of the blocks they store.
4. **Fault Tolerance**: By replicating each block on multiple nodes, HDFS ensures data availability even if some nodes fail.
5. **High Throughput**: HDFS is optimized for streaming large files rather than random access of many small files. It is designed to support large-scale data processing workloads like those typical in big data analytics.

Hadoop Ecosystem Tools Overview

The Hadoop ecosystem comprises a variety of tools and technologies designed to work together for storing, processing, and analyzing large datasets. Some of the key components include:

1. **Hadoop Common**: The common utilities and libraries that support other Hadoop modules.
2. **HDFS (Hadoop Distributed File System)**: As discussed, the primary storage system for Hadoop.
3. **YARN (Yet Another Resource Negotiator)**: Manages resources in the Hadoop cluster and schedules jobs.
4. **MapReduce**: A programming model and processing engine for writing applications that process vast amounts of data in parallel on large clusters.
5. **HBase**: A distributed, scalable, big data store modeled after Google's Bigtable. It provides real-time read/write access to large datasets.
6. **Hive**: A data warehousing and SQL-like query language for Hadoop. It allows for querying and managing large datasets residing in HDFS.
7. **Pig**: A high-level platform for creating programs that run on Hadoop. It provides a scripting language called Pig Latin, which is used for expressing data flows.

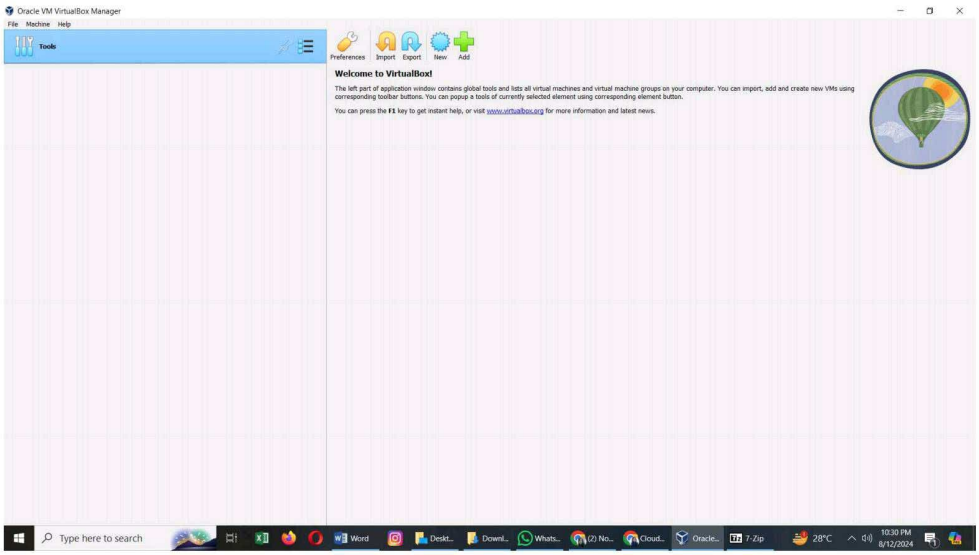
8. **Sqoop**: A tool for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.
9. **Flume**: A distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
10. **Oozie**: A workflow scheduler system to manage Hadoop jobs. It allows users to define a series of actions to be performed and manage their execution.
11. **Zookeeper**: A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.
12. **Ambari**: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters.
13. **Spark**: An open-source, distributed computing system that provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.



-Installing Hadoop.

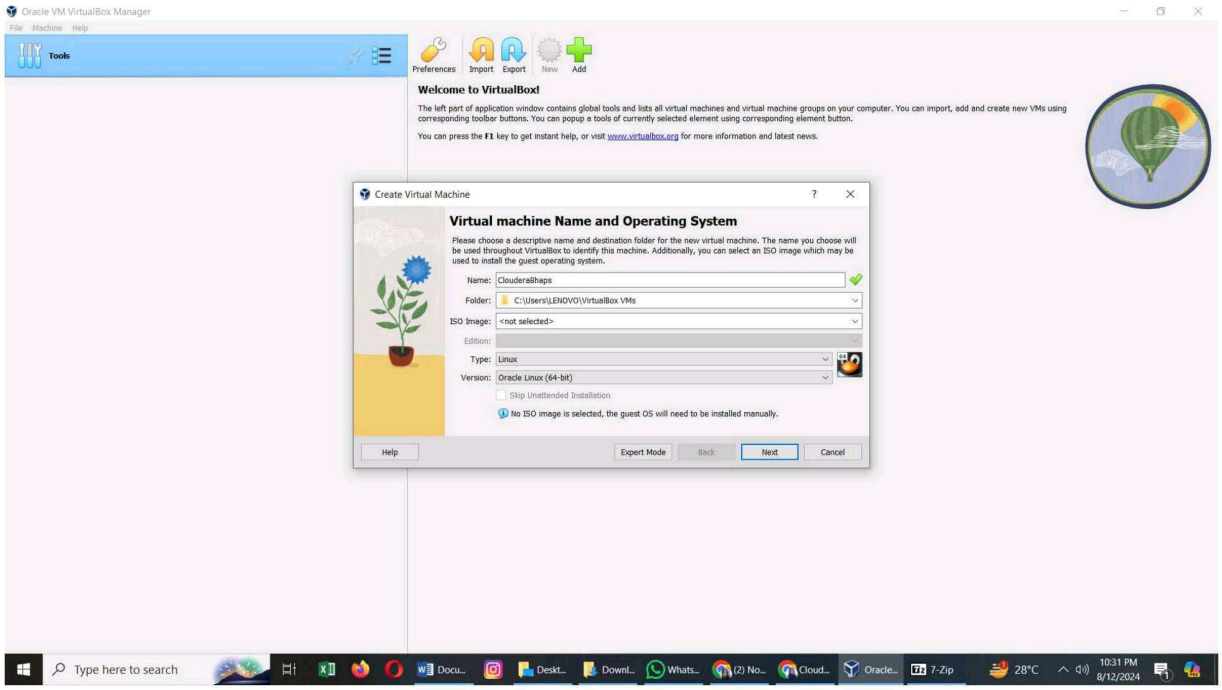
Hadoop Installation using Cloudera Guide

Step1: Open the Oracle Virtual Box:

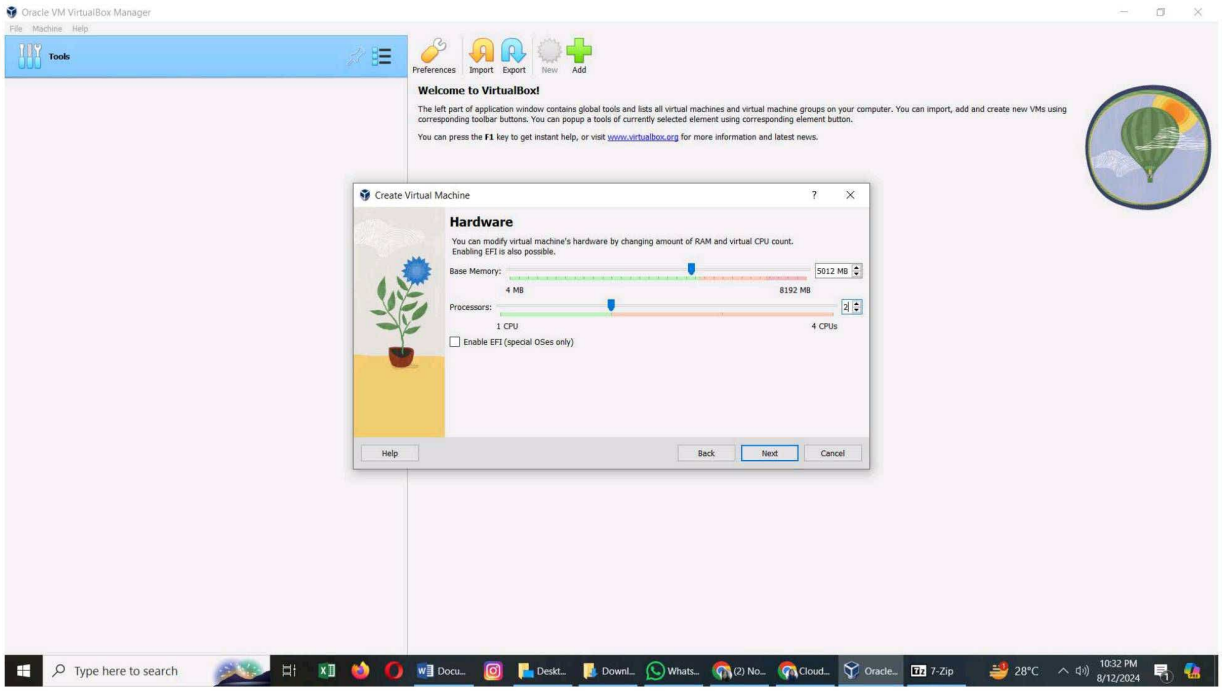


Step2: Click New in the top left corner

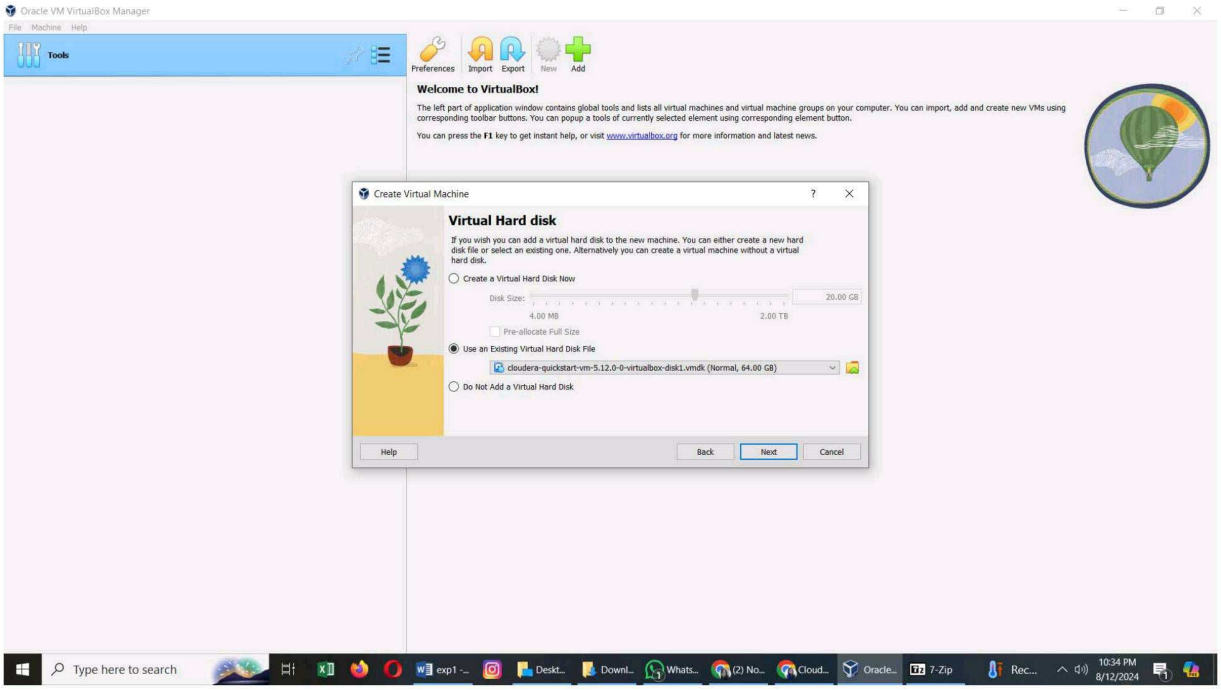
Step3: Give a name for your cloudea virtual machine and select type as 'Linux' and version as 'Other Linux (64-bit) and click Next



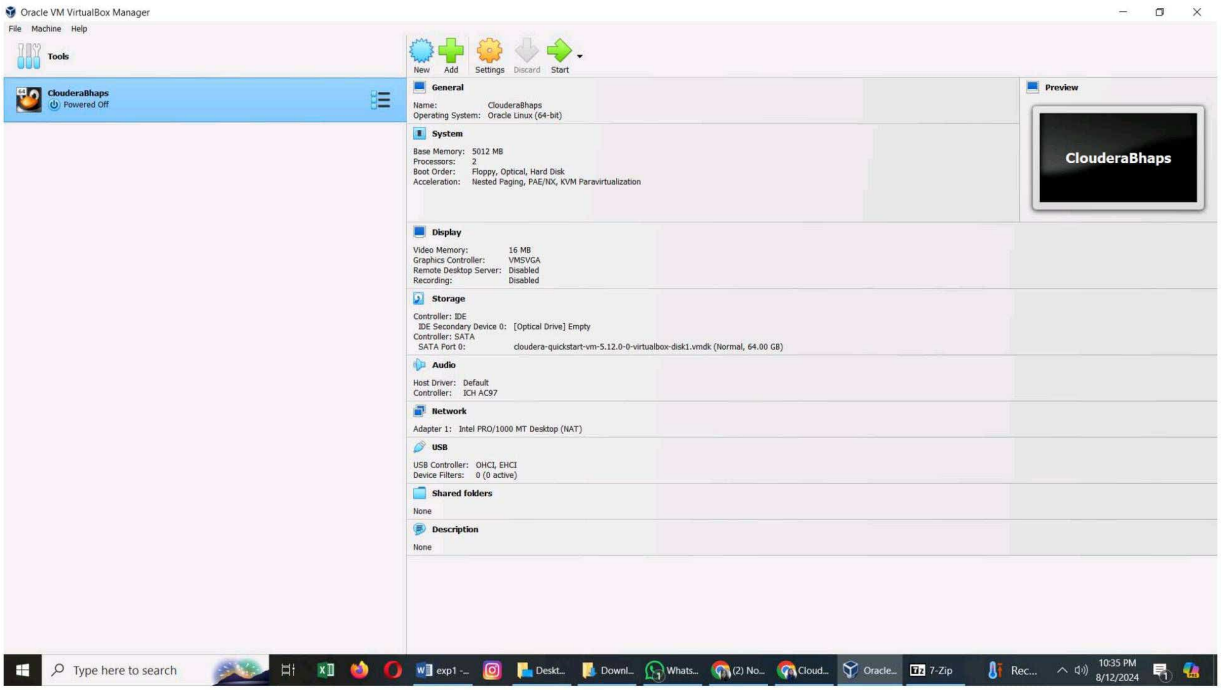
Step4: Give atleast 4096 MB of memory and click Next



Step5: Select the option use an existing virtual hard disk file and click the browse link and then Browse and select the downloaded vmdk file, click open and click on create.



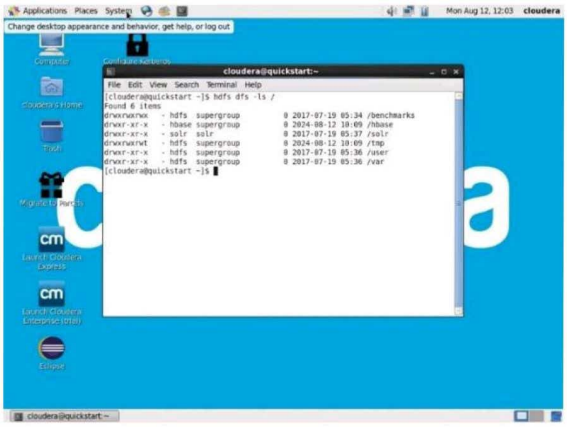
Select the virtual machine and click Start. Wait for all configurations to setup.

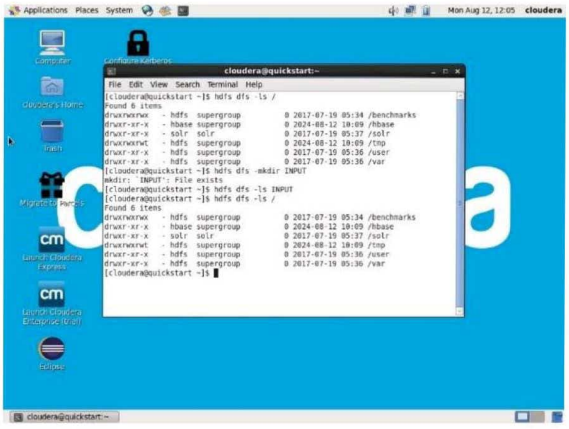


Hadoop Commands:

B	C
Description	Commands
Help	hdfs dfs -help
For listing of files	hdfs dfs -ls /
For making/creating new directory	hdfs dfs -mkdir Input
For listing files in root directory	hdfs dfs -ls Input

Description	Commands
For listing files and directories recursively	[cloudera@quickstart ~]$ hdfs dfs -ls -R /
Copying files from local system into HDFS	[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Desktop/StudentInfo.txt Input/StuentInfo.txt
Retrieving files from HDFS - copies file from HDFS to current working directory.	[cloudera@quickstart ~]$ hdfs dfs -get Input/StudentInfo.txt home/cloudera/Desktop/StudentInfo1.txt
Display file	[cloudera@quickstart ~]$ hdfs dfs -cat Input/StudentInfo.txt
Display head- first few lines of text file	`[cloudera@quickstart ~]$ hdfs dfs-cat Input/StudentInfo.txt
Display tail - last few lines of text file	[cloudera@quickstart ~]$ hdfs dfs-tail Input/StudentInfo.txt
Deleting files from HDFS	[cloudera@quickstart ~]$ hdfs dfs -rm Input/StudentInfo.txt
Deleting files from HDFS	[cloudera@quickstart ~]$ hdfs dfs -rm -r /Input





Conclusion: This experiment demonstrated how to set up and use Hadoop and HDFS on Cloudera. It provided practical experience in managing files in HDFS and running basic Hadoop tasks, reinforcing essential skills for handling big data.