Course Name & Code: Big Data Analytics Lab & CSL702

Experiment No.: 01

Experiment Title: Hadoop HDFS Practical:

A. HDFS Basics, Hadoop Ecosystem Tools Overview
B. Installing Hadoop
C. Copying file to Hadoop
D. Copy from the Hadoop file system and delete a file
E. Moving and displaying files in HDFS
F. Programming exercises on Hadoop

Student Roll No. & Name: Aditya singh Dinesh Kumar Maurya

Date of Performance: **Date of Submission:**

Course /Lab Outcome:

Assessment

Sr. No.	**Parameters for Assessment**	**Marks**	**Rubrics**		
1	**Practical Performance / Active Participation (03 Marks)**		Above Average (03)	Average (02)	Below Average (01)
2	**Report Presentation (02 Marks)**		Above Average (02)	Average (01)	Below Average (00)
3	**Understanding (03 Marks)**		Above Average (03)	Average (02)	Below Average (01)
4	**Regularity in submission (02 Marks)**		Above Average (02)	Average (01)	Below Average (00)

Total Marks (10):

Teacher's Name & Signature Date:

BDA Experiment No.: 01

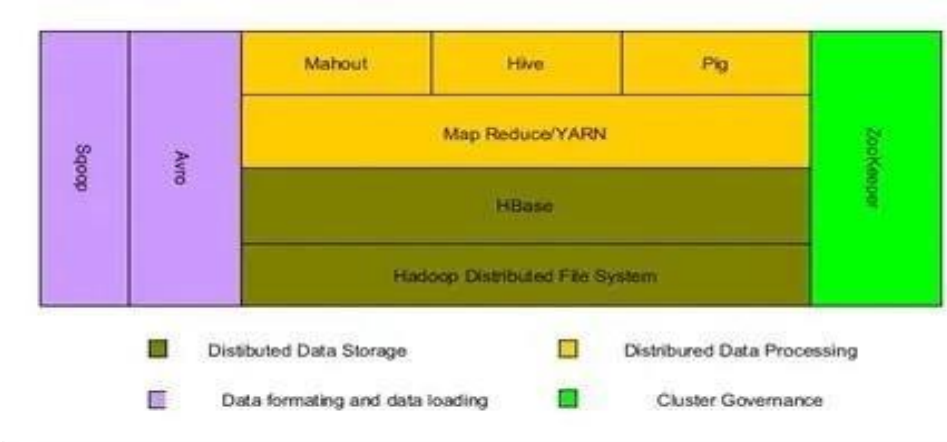
Aim: Hadoop HDFS Practical:

A. DFS Basics, Hadoop Ecosystem Tools Overview
B. Installing Hadoop
C. Copying file to Hadoop
D. Copy from the Hadoop file system and delete a file
E. Moving and displaying files in HDFS
F. Programming exercises on Hadoop **Theory:**

A. HDFS Basics, Hadoop Ecosystem Tools Overview

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 a single server to thousands of machines, each offering local computation and storage.

Hadoop Ecosystem is a platform or a suite that provides various services to solve big data problems. It includes Apache projects and various commercial tools and solutions. There are four major elements of Hadoop i.e. HDFS, MapReduce, YARN, and Hadoop Common. Most of the tools or solutions are used to supplement or support these major elements. All these tools work collectively to provide services such as absorption, analysis, storage, and maintenance of data, etc.



1. **HDFS**

- Hadoop Distributed File System (HDFS) is one of the largest Apache projects and the primary storage system of Hadoop. It employs a NameNode and DataNode architecture. It is a distributed file system able to store large files running over the cluster of commodity hardware.

- Name Node is the prime node that contains metadata (data about data) requiring comparatively fewer resources than the data nodes that store the actual data. These data nodes are commodity hardware in the distributed environment. Undoubtedly, making Hadoop cost-effective.

- HDFS maintains all the coordination between the clusters and hardware, thus working at the heart of the system.

2. **Hive**

- Hive is an ETL and Data warehousing tool used to query or analyze large datasets stored within the Hadoop ecosystem.
- Hive has three main functions: data summarization, query, and analysis of unstructured and semi-structured data in Hadoop. It features a SQL-like interface, HQL language that works similar to SQL and automatically translates queries into MapReduce jobs.

- Hive provides a warehouse structure for other Hadoop input sources and SQL-like access for data in HDFS.

- Hive's query language, HiveQL, compiles to MapReduce and also allows user-defined functions (UDFs). Hive's data model is based primarily on three related data structures: tables, partitions and buckets. Tables correspond to HDFS directories that are divided into partitions, which in turn can be divided into buckets.

3. **Map Reduce**

- This is another data processing layer of Hadoop. It has the capability to process large structured and unstructured data as well as to manage very large data files in parallel by dividing the job into a set of independent tasks (sub-job).

- MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:

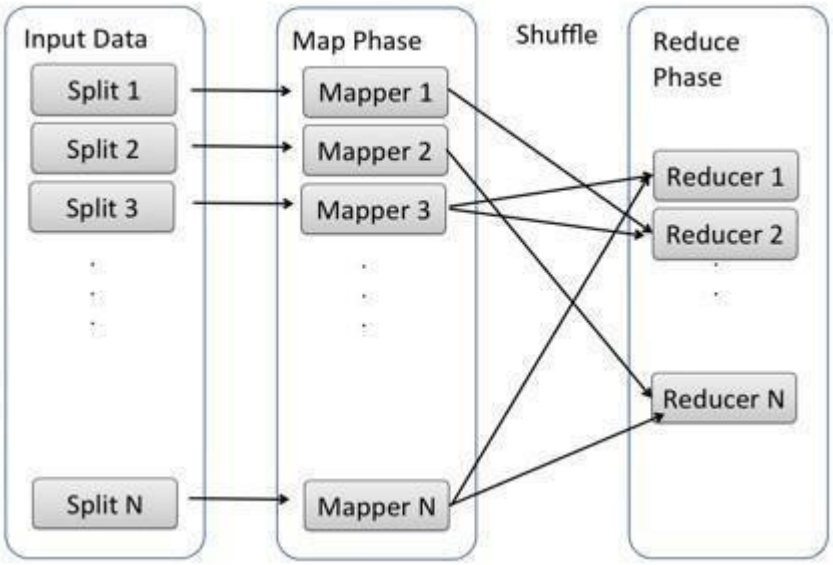
(a) Map() performs sorting and filtering of data and thereby organizes them in the form of a group. Map generates a key-value pair-based result which is later on processed by the Reduce() method.

(b) Reduce(), as the name suggests does the summarization by aggregating the mapped data. In simple, Reduce() takes the output generated by Map() as input and combines those tuples into a smaller set of tuples.

- Main Components of MapReduce

The main components of MapReduce are listed below:

1. JobTrackers: JobTracker is the master which manages the jobs and resources in the cluster. The JobTracker tries to schedule each map on the TaskTracker which is running on the same DataNode as the underlying block.
2. TaskTrackers: TaskTrackers are slaves which are deployed on each machine in the cluster. They are responsible for running the map and reduce tasks as instructed by the JobTracker.
3. JobHistoryServer: JobHistoryServer is a daemon that saves historical information about completed tasks/applications.



4. **YARN**

- Yet Another Resource Negotiator, as the name implies, YARN is the one who helps to manage the resources across the clusters. In short, it performs scheduling and resource allocation for the Hadoop System.
- Consists of three major components i.e.

Resource Manager

Nodes Manager

Application Manager

- Resource manager has the privilege of allocating resources for the applications in a system whereas Node managers work on the allocation of resources such as CPU, memory, bandwidth per machine and later on acknowledges the resource manager. Application manager works as an interface between the resource manager and node manager and performs negotiations as per the requirement of the two.

5. **PIG**

- This is a high-level scripting language used to execute queries for larger datasets that are used within Hadoop.
- Pig was basically developed by Yahoo which works on a pig Latin language, which is Query based language similar to SQL. It is a platform for structuring the data flow, processing and analyzing huge data sets.
- Pig does the work of executing commands and in the background, all the activities of MapReduce are taken care of. After the processing, pig stores the result in HDFS. Pig Latin language is specially designed for this framework which runs on Pig Runtime. Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

6. **Spark**

- Apache Spark is a fast, in-memory data processing engine suitable for use in a wide range of circumstances.
- Spark can be deployed in several ways, it features Java, Python, Scala, and R programming languages, and supports SQL, streaming data, machine learning, and graph processing, which can be used together in an application.
- It's a platform that handles all the process consumptive tasks like batch processing, interactive or iterative real-time processing, graph conversions, and visualization, etc.

7. **HBase**

- HBase "is an open-source, distributed, versioned, column-oriented store" that sits on top of HDFS. HBase is based on Google's Bigtable. HBase is based on columns rather than rows.
- It's a NoSQL database which supports all kinds of data and thus capable of handling anything of Hadoop Database. It provides capabilities of
  Google's BigTable, thus able to work on Big Data sets effectively.
- At times where we need to search or retrieve the occurrences of something small in a huge database, the request must be processed within a short quick span of time. At such times, HBase comes handy as it gives us a tolerant way of storing limited data.

8. Sqoop

- Sqoop ("SQL-to-Hadoop") is a tool which transfers data in both ways between relational systems and HDFS or other Hadoop data stores such as Hive or HBase. Sqoop can be used to import data from external structured databases into HDFS or any other related systems such as Hive and HBase.
- On the other hand, Sqoop can also be used to extract data from Hadoop and export it to external structured databases such as relational databases and enterprise data warehouses.

9. Zookeeper

- There was a huge issue of management of coordination and synchronization among the resources or the components of Hadoop which resulted in inconsistency, often.
- Zookeeper overcame all the problems by performing synchronization, intercomponent based communication, grouping, and maintenance.

10. **Oozie**

- Oozie is a job coordinator and workflow manager for jobs executed in Hadoop. It is integrated with the rest of the Apache Hadoop stack.
- It supports several types of Hadoop jobs, such as Java map-reduce, Streaming mapreduce, Pig, Hive and Sqoop as well as system-specific jobs such as Java programs and shell scripts.
- An Oozie workflow is a collection of actions and Hadoop jobs arranged in a Directed Acyclic Graph (DAG), since tasks are executed in a sequence and also are subject to certain constraints.

B. Installing Hadoop

In this experiment, we are going to install Cloudera for Hadoop by seting up a virtual environment for running Hadoop MapReduce code, run simple commands, create a file, search through the file.

Setup Hadoop Virtual Environment

To setup a single node cluster, on your laptop, you need following software:

1. Oracle Virtual Box – software to install one or more virtual machines.

Get this software from https://www.virtualbox.org/

2. Cloudera VM – this software acts as a virtual machine. By default, Cloudera consist of packages of most of the cloud computing frameworks like Hadoop, Spark, Hive, Pig, etc. Download Cloudera QuickStart VM for Virtual Box platform. Go get this software, go to http://www.cloudera.com/ ☾ Select Get Started ☾ Select

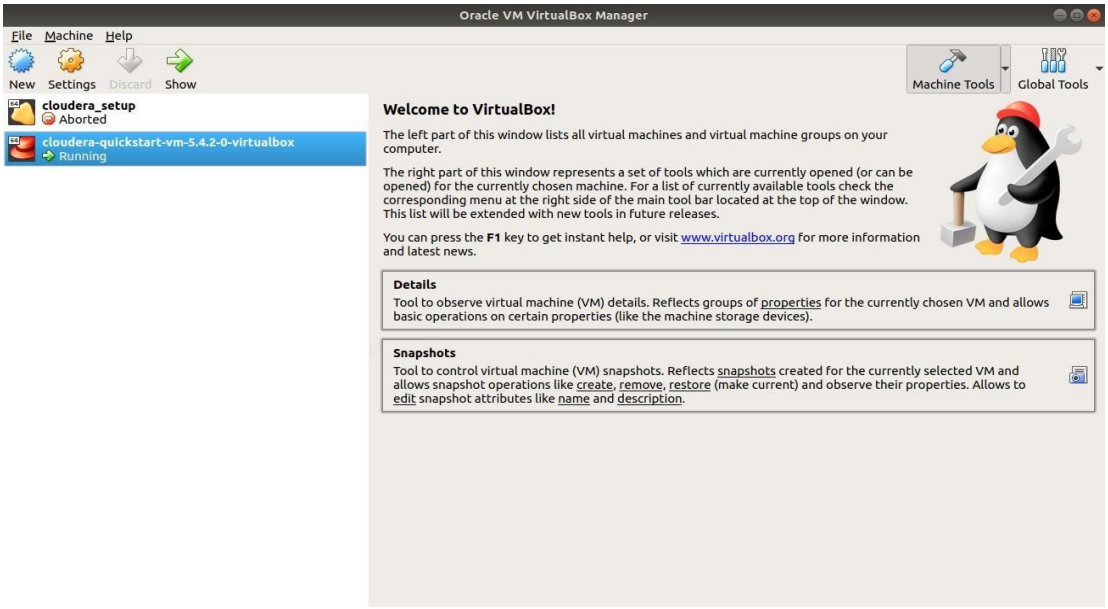
Try Now ☾ Select
Download Now inside QuickStarts box ☾ Select a platform as Virtual Box.

Fill a form that pops up after this step and download Cloudera. (or)

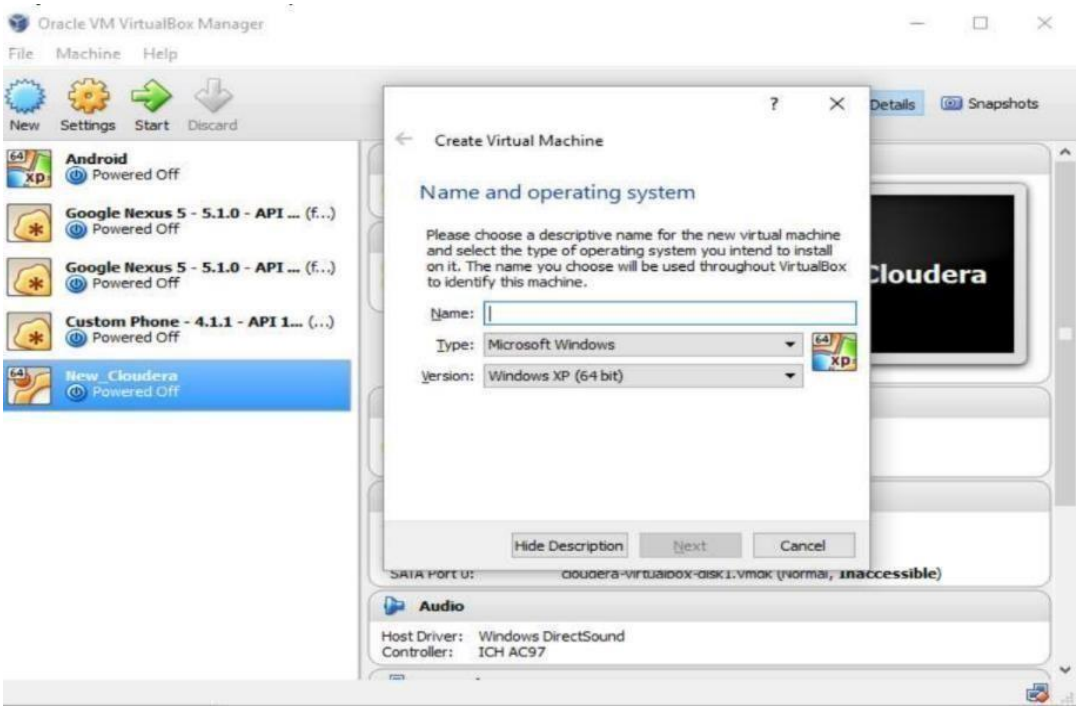
You can install the Cloudera from following link:
https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-v m-5.12.0-0-virtualbox.zip

Please follow the below steps to import the cloudera vmdk file to your local machine through

Virtual Box or you can follow a video from the following link to install Cloudera and Virtual Box:
https://www.youtube.com/watch?v=BeCtjd86 YXo Step1: Open the Oracle Virtual Box:



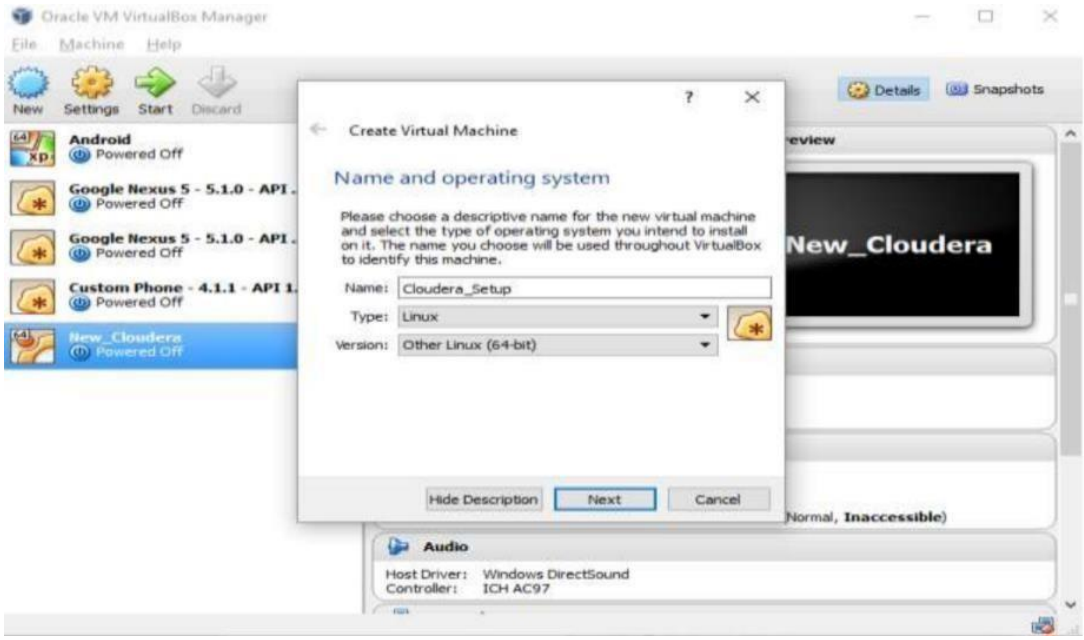
Step2: Click New in the top left corner



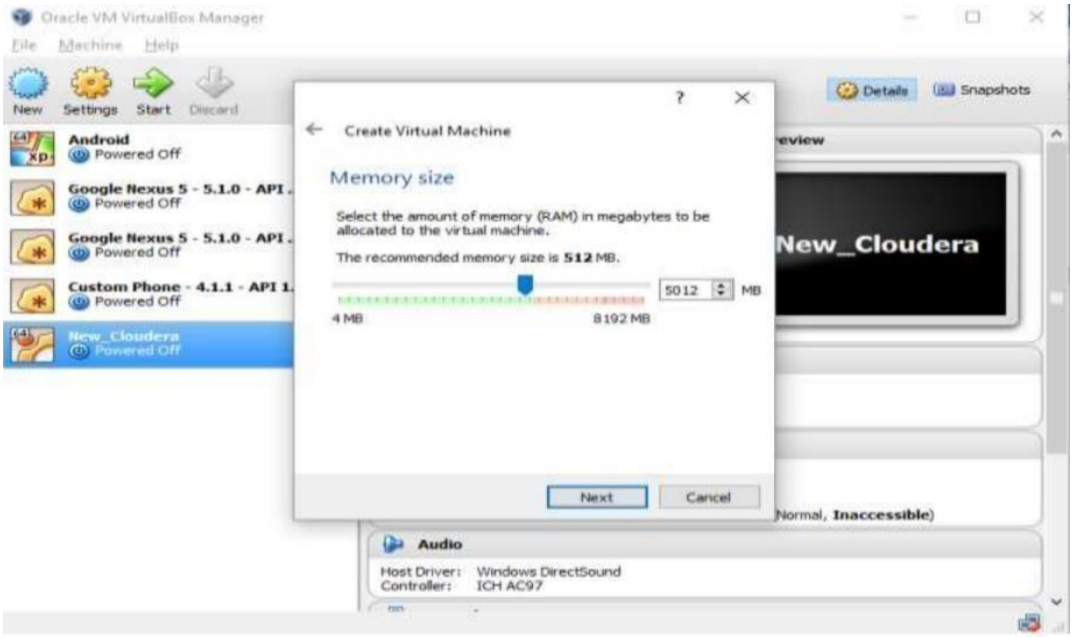
Step3: Give a name for your cloudera virtual machine and select type as

'Linux' and version as

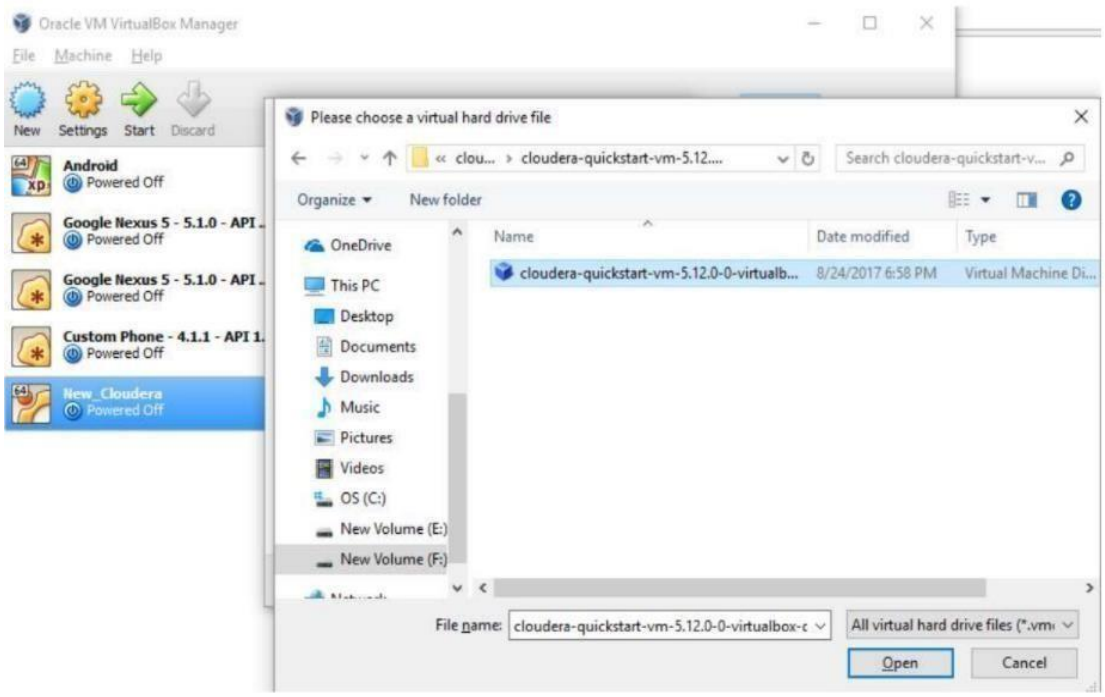
'Other Linux (64-bit) and click Next



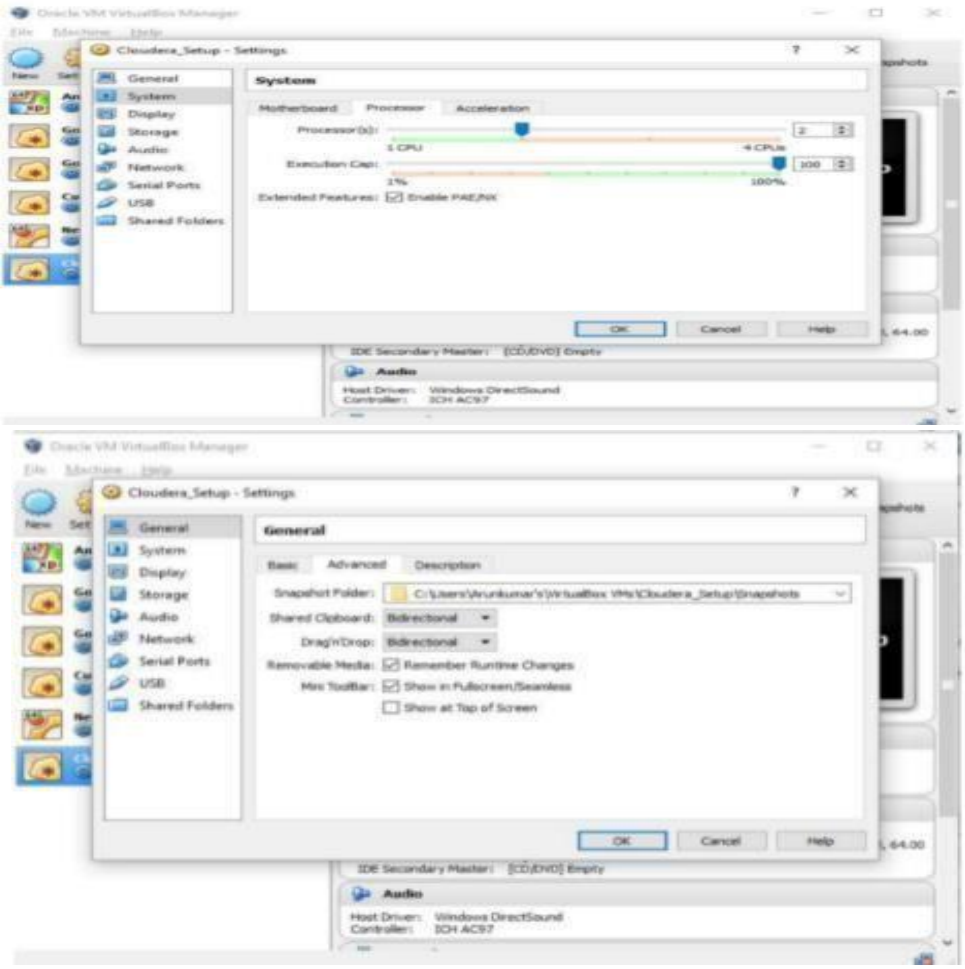
Step4: Give atleast 4096 MB of memory and click Next



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.



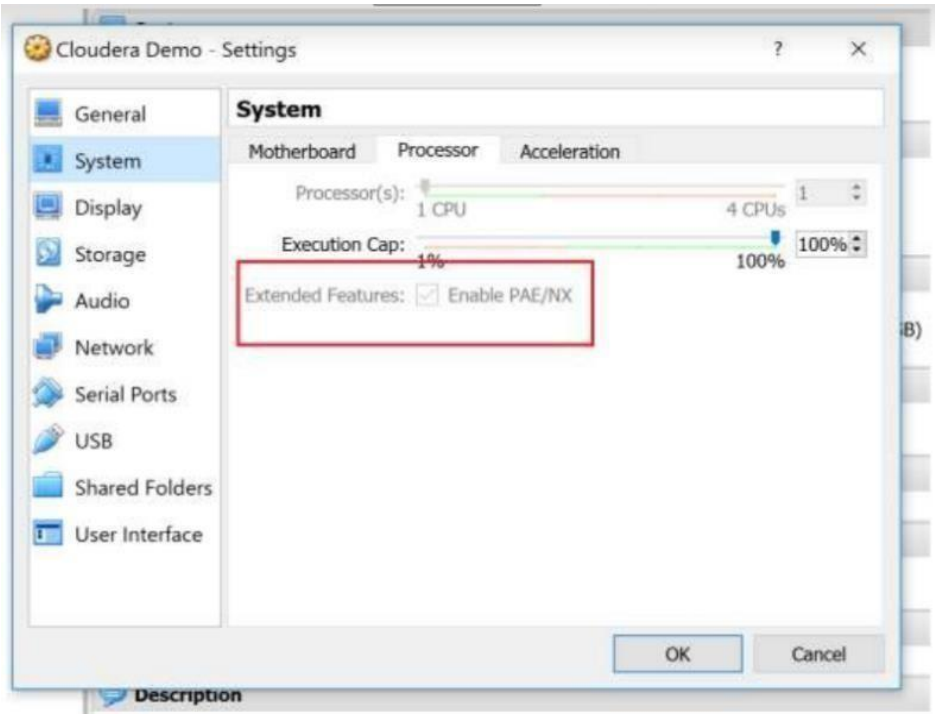
Step6: Ensure the settings are similar to the below screenshots



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

Fixing BIOS problem while installing Cloudera:

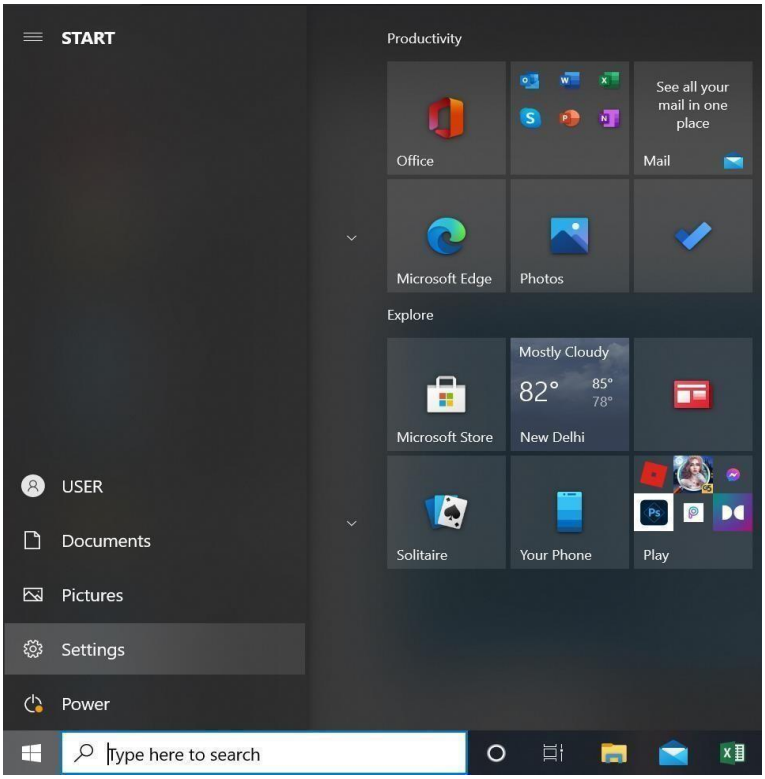
1. In case of BIOS error in Windows environment enable PAE/NX, and check if issue is resolved.



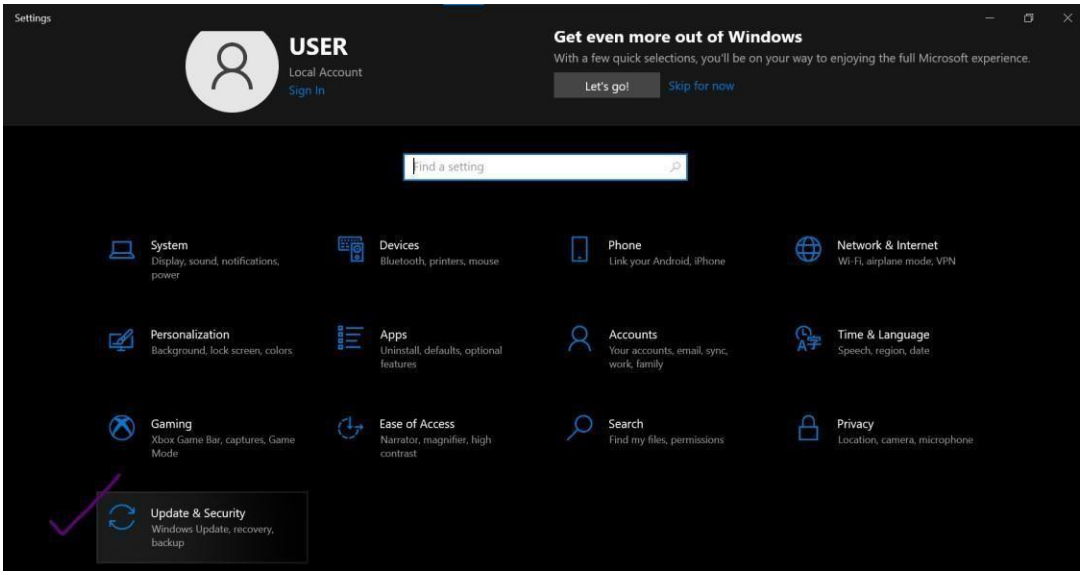
2. If issue still exists, Restart computer go to bios settings and enable virtualization technology as shown in below video. https://www.youtube.com/watch?v=-Wa7TGjmn5M

To access BIOS settings in Windows 10

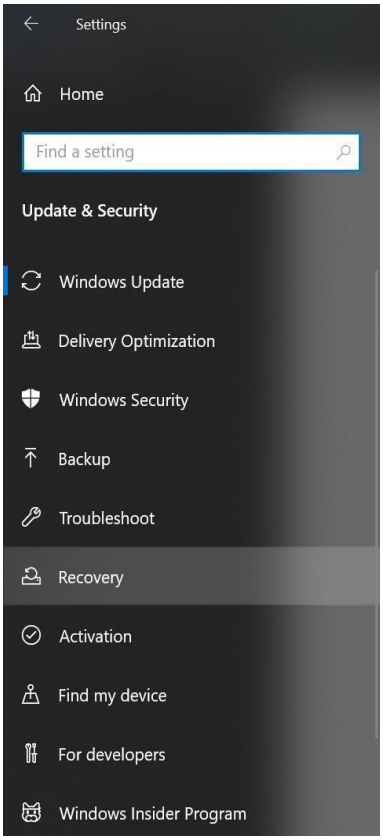
1. Navigate to settings. You can get there by clicking the gear icon on the Start menu.



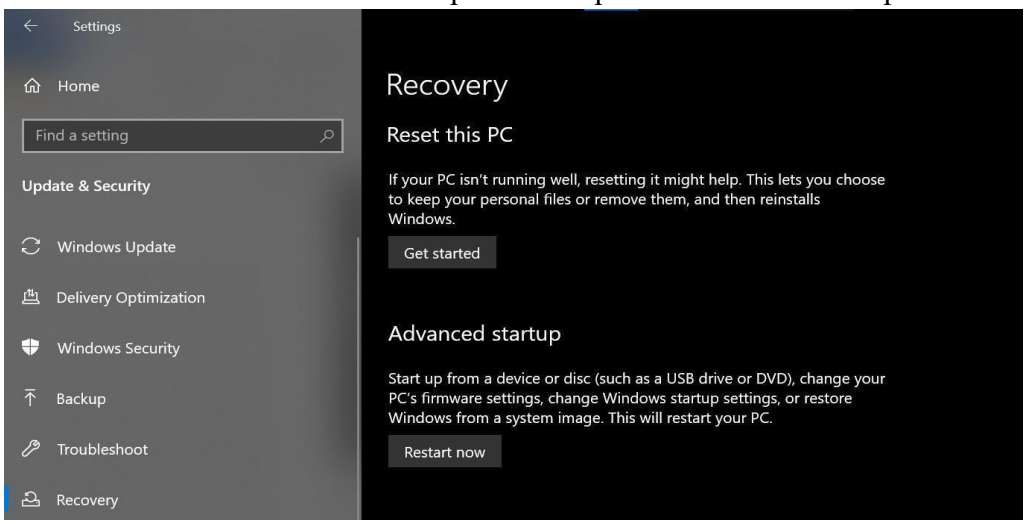
2. Select Update & security.



3. Select Recovery from the left menu.



4. Click Restart Now under Advanced startup. The comptuer will reboot to a special menu.



5. Click Troubleshoot.

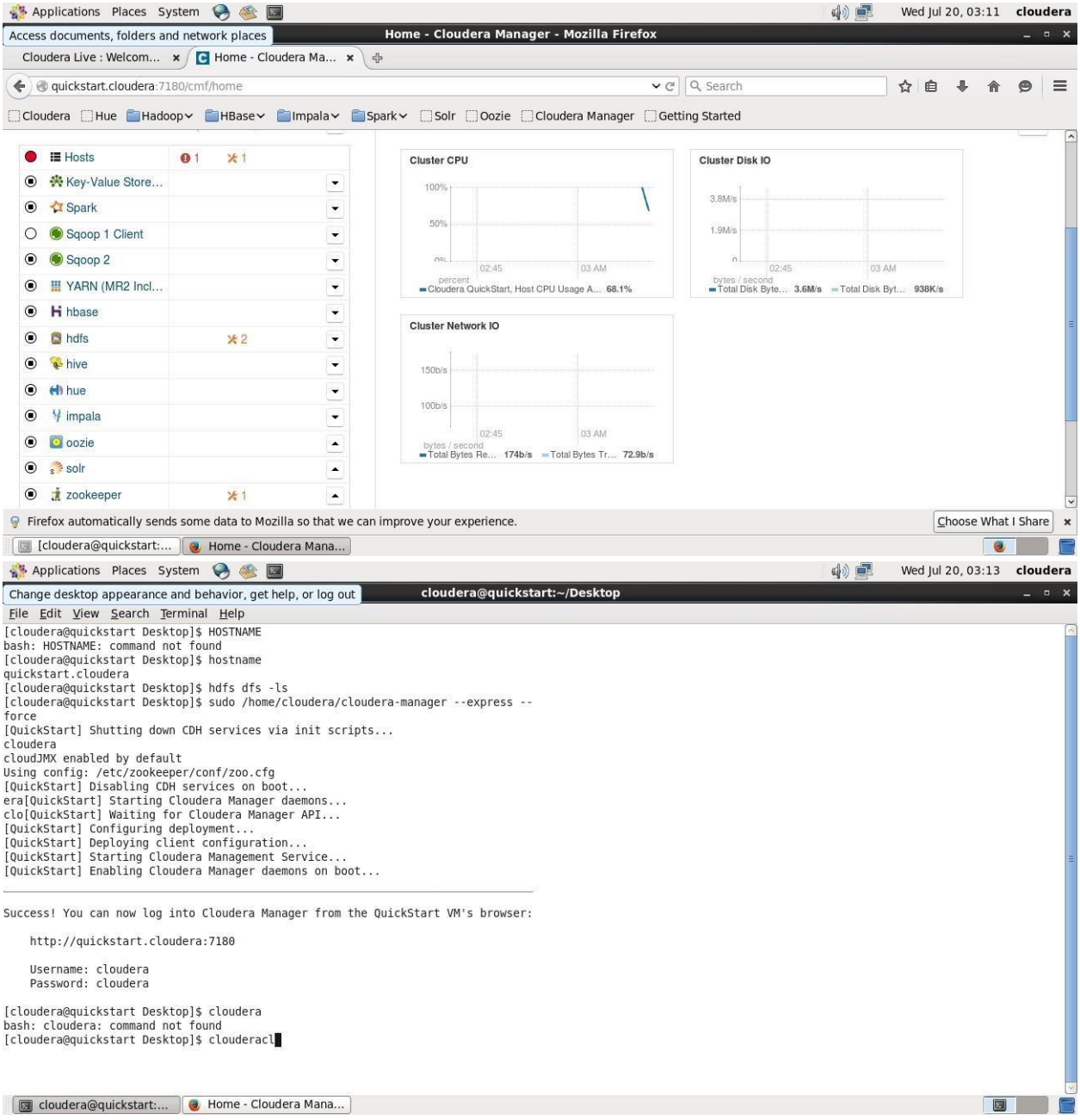
6. Click Advanced options.

7. Select UEFI Firmware Settings.

8. Click Restart.

Your system will restart and take you to the BIOS.

We have successfully installed Hadoop in our system.



C. Copying file to Hadoop HDFS commands

Sr No.	Description	Commands
1	Help	[cloudera@quickstart~]$ hdfs dfs -help ls
2	For listing of files	[cloudera@quickstart~]$ hdfs dfs –ls/
3	For making/creating new directory	[cloudera@quickstart~]$ hdfs dfs -mkdir myfile
4	For listing directory in root directory	[cloudera@quickstart~]$ hdfs dfs -ls myfile
5	For listing files and directories recursively	[cloudera@quickstart~]$ hdfs dfs -ls -R /
6	Copying files from local system into HDFS	[cloudera@quickstart~]$ hdfs dfs -put /home/cloudera/Desktop/newfile.txt' /myfile/newfile.txt
7	Retriving files from HDFSCopies file from HDFS to current working directory	[cloudera@quickstart~]$ hdfs dfs -get myfile/newfile.txt /home/cloudera/Desktop/newfile.txt'
		[cloudera@quickstart~]$ hdfs dfs -cat myfile/newfile.txt
		[cloudera@quickstart~]$ hdfs dfs -cat myfile/newfile.txt |home
		[cloudera@quickstart~]$ hdfs dfs -tail myfile/newfile.txt

8	Deleting files from HDFS	[cloudera@quickstart~]$ hdfs dfs -rm myfile/newfile.txt [cloudera@quickstart~]$ hdfs dfs –rm /myfil

D. Copy from Hadoop File System and deleting file

```
[cloudera@quickstart ~]$ hadoop fs -rm /mydoc/newfile.txt
Deleted /mydoc/newfile.txt
[cloudera@quickstart ~]$ hadoop fs -rmdir /mydoc
[cloudera@quickstart ~]$ hadoop fs -ls /
Found 6 items
drwxrwxrwx   - hdfs  supergroup          0 2017-07-19 05:34 /benchmarks
drwxr-xr-x   - hbase supergroup          0 2021-09-19 01:41 /hbase
drwxr-xr-x   - solr  solr                0 2017-07-19 05:37 /solr
drwxrwxrwt   - hdfs  supergroup          0 2021-09-19 01:41 /tmp
drwxr-xr-x   - hdfs  supergroup          0 2017-07-19 05:36 /user
drwxr-xr-x   - hdfs  supergroup          0 2017-07-19 05:36 /var
[cloudera@quickstart ~]$
```

E. Moving and displaying file in HDFS.

```
[cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Name.txt /MyName
[cloudera@quickstart ~]$ hadoop fs -ls /
Found 9 items
drwxr-xr-x   - cloudera supergroup          0 2022-09-25 23:59 /MyName
drwxrwxrwx   - hdfs     supergroup          0 2017-07-19 05:34 /benchmarks
drwxr-xr-x   - hbase    supergroup          0 2022-09-25 23:15 /hbase
drwxr-xr-x   - cloudera supergroup          0 2022-09-25 23:41 /input1
drwxr-xr-x   - cloudera supergroup          0 2022-09-25 23:44 /out1
drwxr-xr-x   - solr     solr                0 2017-07-19 05:37 /solr
drwxrwxrwt   - hdfs     supergroup          0 2022-08-17 02:17 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2017-07-19 05:36 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-07-19 05:36 /var
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ hadoop fs -mkdir /MyName
[cloudera@quickstart ~]$ cat > Name.txt
Name : Omkar
```

```
cloudera@quickstart ~]$ hadoop fs -put /home/cloudera/Name.txt /MyName
cloudera@quickstart ~]$
```