

MGM’s

Jawaharlal Nehru Engineering College Aurangabad

Affiliated to Dr.B.A.TechnologicalUniversity ,Lonere Maharashtra

ISO 9001:2015,140001:2015Certified,AICTE Approved

**Department of Computer Science & Engineering**

**LAB MANUAL**

Programme(UG/PG) : UG

Year : Final Year

Semester : VIII

Course Code : BTCOL707

Course Title : Big Data Analytics Laboratory

Prepared By

Mr. S. N. Jaiswal

Assistant Professor

Department of Computer Science & Engineering

**FOREWORD**

It is my great pleasure to present this laboratory manual for Final year engineering students for the subject of System Administration.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and e­xactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9001:2015,140001:2015 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr. H. H. Shinde

Principal

**LABORATORY MANUAL CONTENTS**

This manual is intended for the Final year students of Computer Science & Engineering in the subject of Big Data Analytics. This manual typically contains practical/Lab Sessions related to Big Data Analytics covering various aspects related the subject to enhanced understanding.

Big Data Analytics provide students the idea of managing big data, analyzing it and carry out meaningful information from big data. The Goal of Big Data Analytics is to learn, use and develop simple application using Apache Hadoop ecosystem..

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

**Mr.S.N.Jaiswal Dr. Vijaya Musande**

HOD

Subject Teacher

**LIST OF EXPERIMENTS**

Course Code: BTCOL707

Course Title: Big Data Analytics Laboratory

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name of the Experiment** | **Page No** |
| 1. | Basic CRUD operations in MongoDB. Design a sample database for any application |  |
| 2. | Installation of Cassandra and hands-on practice of it. |  |
| 3. | Cloudera setup and Installation using VM |  |
| 4. | Setup and installation of Apache Hadoop. |  |
| 5. | File management task in Apache Hadoop |  |
| 6. | Map Reduce Paradigm |  |
| 7. | Install, Deploy and configure Apache Spark. Develop application using Apache Spark. |  |
| 8. | Application development using Apache PySpark |  |
| 9. | Install and configure Apache Kafka. |  |
| 10. | Integrate of Apache kafka with Apache Spark |  |
| 11. | Data Analytics using Apache Spark on Amazon food dataset. |  |
| 12. | Classification, clustering and frequent item set using Spark MLLib. |  |
| 13. | Installation and use of Hbase. |  |
| 14. | Design of simple project like tweet analysis, fraud detection or any other using different components of Apache Hadoop. |  |

**DOs and DON’Ts in Laboratory:**

1. Make entry in the Log Book as soon as you enter the Laboratory.

2. All the students should sit according to their roll numbers starting from their left to right.

3. All the students are supposed to enter the terminal number in the log book.

4. Do not change the terminal on which you are working.

5. All the students are expected to get at least the algorithm of the program/concept to be implemented.

6. Strictly observe the instructions given by the teacher/Lab Instructor.

7. Do not disturb machine Hardware / Software Setup.

**Instruction for Laboratory Teachers:**

1. Submission related to whatever lab work has been completed should be done during the next lab session along with signing the index.

2. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

3. Continuous assessment in the prescribed format must be followed.

**MGM’s**

****

**Jawaharlal Nehru Engineering College, Aurangabad**

**Department of Computer Science and Engineering**

**Vision of CSE Department**

To develop computer engineers with necessary analytical ability and human values who can creatively design, implement a wide spectrum of computer systems for welfare of the society.

**Mission of the CSE Department:**

1. Preparing graduates to work on multidisciplinary platforms associated with their professional position both independently and in a team environment.
2. Preparing graduates for higher education and research in computer science and engineering enabling them to develop systems for society development.

**Programme Educational Objectives**

**Graduates will be able to**

1. To analyze, design and provide optimal solution for Computer Science & Engineering and multidisciplinary problems.
2. To pursue higher studies and research by applying knowledge of mathematics and fundamentals of computer science.
3. To exhibit professionalism, communication skills and adapt to current trends by engaging in lifelong learning.

**Programme Outcomes (POs):**

**Engineering Graduates will be able to:**

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems anddesign system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change.

**LABORATORY OUTCOMES**

The practical/exercises in this section are psychomotor domain Learning Outcomes (i.e. subcomponents of the COs), to be developed and assessed to lead to the attainment of the competency.

**LO-1:** Setup the Hadoop/Cloudera framework

**LO-2:** Perform file management task in Hadoop

**LO-3:** Devise simple applications using Map Reduce API and Java/Python

**LO-4:** Implement CRUD operations in MongoDB / Cassandra

**LO-5:** Develop simple streaming applications using Apache Kafka

**LO-6:** Analyze the data using Spark Mlib

**1. Lab Exercise**

Exercise No 1: (2 Hours) – 1 Practical

# Aim: -Installations of mongoDB on windows/linux operating system

**Objectives:**

1. Student will able to install mongoDB on either Wnidows or Linux operating system.

2. Students should able to design database for any application using mongoDBand perform CRUD operations on it.

3. Student will able to use mongoDB through Java / Python or any other programming framework.

**THEORY:**

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability.MongoDB works on concept of collection and document.MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++.

Database is a physical container for collections. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose.

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data

Sample Document

Following example shows the document structure of a blog site, which is simply a comma separated key value pair

{

\_id:ObjectId(7df78ad8902c)

title:'MongoDB Overview',

description:'MongoDB is no sql database',

by:'Open Source Community’,

url: ‘https://www.mongodb.com/',

tags:['mongodb','database','NoSQL'],

likes:100,

comments:[

{

user:'user1',

message:'My first comment',

dateCreated:newDate(2011,1,20,2,15),

like:0

},

{

user:'user2',

message:'My second comments',

dateCreated:newDate(2011,1,25,7,45),

like:5

}

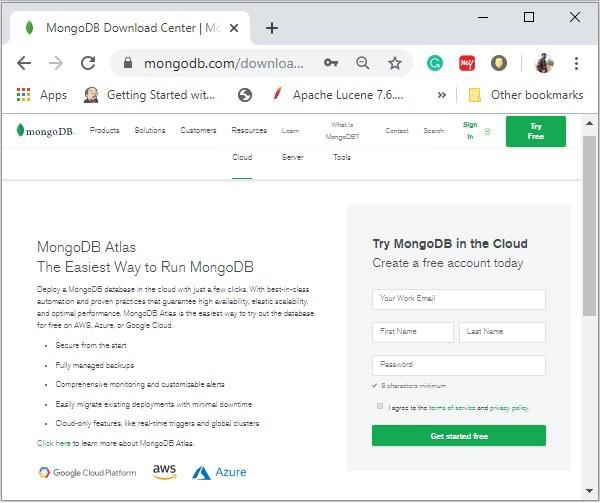
]

}

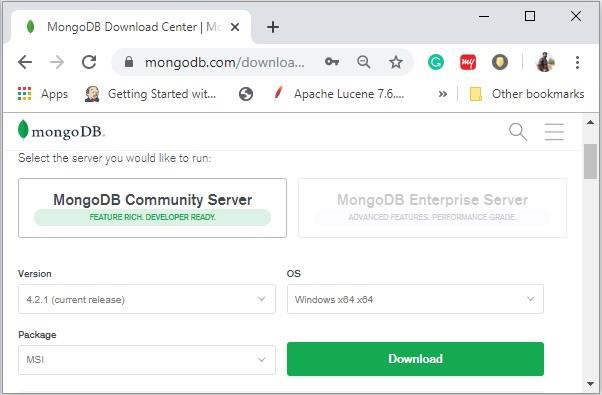
**\_id** is a 12 bytes hexadecimal number which assures the uniqueness of every document. You can provide \_id while inserting the document. If you don’t provide then MongoDB provides a unique id for every document. These 12 bytes first 4 bytes for the current timestamp, next 3 bytes for machine id, next 2 bytes for process id of MongoDB server and remaining 3 bytes are simple incremental VALUE.

**Installation of mongoDB on Windows system:**

To install MongoDB on Windows, first download the latest release of MongoDB from  [https://www.mongodb.com/download-center](https://www.mongodb.com/download-center" \t "_blank)



Enter the required details, select the ***Server*** tab, in it you can choose the version of MongoDB, operating system and, packaging as:



Now install the downloaded file, by default, it will be installed in the folder **C:\Program Files\**.

MongoDB requires a data folder to store its files. The default location for the MongoDB data directory is c:\data\db. So you need to create this folder using the Command Prompt.

Then you need to specify set the **dbpath** to the created directory in **mongod.exe**. For the same, issue the following commands.

In the command prompt, navigate to the bin directory current in the MongoDB installation folder. Suppose my installation folder is **C:\Program Files\MongoDB.** Run the commandas***mongod.exe --dbpath "C:\data"***

This will show **waiting for connections** message on the console output, which indicates that the mongod.exe process is running successfully.

Now to run the MongoDB, you need to open another command prompt and issue the following command on command prompt

C:\Program Files\MongoDB\Server\4.2\bin>***mongo.exe***

System shows MongoDB prompt ***>***

**MongoDB : Basic Commands**

* db.help() List of commands
* db.stats() Statistics of MongoDB
* use DATABASE\_NAME create database
  + use myDb create database with name myDb
  + db Check the current database
* show dbs Show the list of database in system

**Inserting record/document in database**

* db.collection.insertOne(….)
* db.collection.insertMany(…)
* db.collection.insert()

***db.student.insert(***

***{***

***name: "anilkumar",***

***roll: 406175,***

***dob: "15 Aug 1999",***

***hobby: ["reading", "Swimming"],***

***address :***

***{area: "Gulmandi", city: “Aurangabad", pin: 431001}***

***}***

***)***

**Outcome:**

To learn the installation process of mongoDB and perform basic operations on database and collections.

**CONCLUSIONS:**

By following above Steps students will able to install mongoDB on Windows. Students will be able to develop schema for specified application using mongoDB and perform basic operations on it.

**Additional Task :**

1. **Install mongoDB on Linux operating system**
2. **Design user interface for mongoDB using either Java or Python.**

**Journal Write-up**

* **Introduction**
* **NoSQL concept**
* **Types of NoSQL**
* **History and Terminology of mongoDB**
* **MongoDB Basic commands**
* **Design a database schema for any application of your choice**
* **Solve minimum 25 queries which includes all aspect of mongoDB**
* **MongoDB connectivity with Java & Python**
* **Conclusion**

**2. Lab Exercise**

Exercise No 2: (2 Hours) – 1 Practical

# Aim: -Installation of Cassandra and hands-on practice of it

**Objectives:**

1. Student will able to install Cassandra

2. Students should able to perform operations by using different commands.

3. Student will able to use different data types

**THEORY:**

Cassandra is a distributed database from Apache. Highly scalable and designed to manage very large amounts of structured data.Provides high availability with no single point of failure. Created at Facebook, it differs sharply from relational database management systems. It was open-sourced by Facebook in July 2008. Cassandra was accepted into Apache Incubator in March 2009. It was made an Apache top-level project since February 2010

It is a column-oriented database. Its distribution design is based on Amazon’s Dynamo and its data model on Google’s Bigtable. Cassandra implements a Dynamo-style replication model with no single point of failure, but adds a more powerful “column family” data model. Cassandra is being used by some of the biggest companies such as Facebook, Twitter, Cisco, Rackspace, ebay, Twitter, Netflix, and more.

Cassandra is an open-source distributed database software for handling NoSQL databases. CQL (Cassandra Query Language) is used. CQL keeps data in tables arranged in a set of rows with columns that 6key-value pairs.CQL tables are grouped in data containers called keyspaces in Cassandra. Data stored in one keyspace is unrelated to other data in the cluster.

**Features of Cassandra:**

1. Elastic scalability − it allows to add more hardware to accommodate more customers and more data as per requirement.
2. Always on architecture − Cassandra has no single point of failure
3. Fast linear-scale performance − Cassandra is linearly scalable
4. Flexible data storage − accommodates all data formats
5. Easy data distribution − provides the flexibility to distribute data where you need by replicating data across multiple data centers.
6. Transaction support − supports ACID properties.
7. Fast writes − It performs fast writes and can store hundreds of terabytes of data.

**Installation steps to be followed:(Windows)**

1. Download and Install Java 8 and set environment variables.
2. Download and install Python 2.7 and set environment variables.
3. Download and Extract Cassandra tar.gz Folder
4. Set CASSANDRA\_HOME & add bin folder to path
5. Start Cassandra from Windows CMD
6. Access Cassandra cqlsh from Windows CMD

**Apache Cassandra Data Types**

1. Built-in data types
2. Collection data types
3. User-defined data types
4. **Built In Data Types:**

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Constants** | **Description** |
| Ascii | Strings | includes character encoding used for strings. |
| Boolean | Booleans | Stored as 16-bit numbers |
| Blob | Blobs | “Binary Large Object” and it is utilized for storing binary data. |
| decimal | integers, floats | Offers precision & scale |
| double | Integers | It represents a 64-bit floating point |
| Float | integers, floats | The FLOAT data type stores decimal point values. |
| Int | Integers | used to store 32-bit signed integers. |
| smallint | Integers | stores 16-bit signed integers. |
| bigint | Integers | BIGINT stores 64-bit signed integers. |
| Text | Strings | use TEXT data types used for text data, represented in UTF8 encoded strings. |
| varchar | Strings | Use VARCHAR for variables or arbitrary characters |
| Inet | Strings | Use it to save and manage IP addresses since it supports both numeric and character representation. |
| counter | Integers | This data type supports two operations: incrementing and decrementing |
| Time | integers, strings | You can store time values in the following format: hh:mm:ss using the time data type. |
| Date | integers, strings | store date values in the format: YYYY-MM-DD. |
| timestamp | integers, strings | combination of time & data |

1. **Collection data types**

|  |  |
| --- | --- |
| Maps | Cassandra can store data in sets of key-value pairs using the *Map* data type. |
| Sets | You can store multiple unique values, using the *Set* data type. |
| Lists | If you need to store multiple values in a specific order, you can use the *List* data type. |

1. **User Defined Data types**

User-Defineddata types (UDTs) allows to create your own data type based on the requirements you need.A UDT consists of multiple data fields of any data type inside a single column. Once you create your user-defined data type, you can change or even remove the fields inside of it

**Shell Commands:**

1. HELP − Displays help topics for all cqlsh commands.
2. CAPTURE − Captures the output of a command and adds it to a file.
3. CONSISTENCY − Shows the current consistency level, or sets a new consistency level.
4. COPY − Copies data to and from Cassandra.
5. DESCRIBE − Describes the current cluster of Cassandra and its objects.
6. EXPAND − Expands the output of a query vertically.
7. EXIT − Using this command, you can terminate cqlsh.
8. PAGING − Enables or disables query paging.
9. SHOW − Displays the details of current cqlsh session such as Cassandra version, host, or data type assumptions.
10. SOURCE − Executes a file that contains CQL statements.
11. TRACING − Enables or disables request tracing.

* **Create keyspace**

*Syntax:*

*Create keyspace*KeyspaceName*with replication*={*'class*':strategy name, '*replication\_factor*': No of replications on different nodes};

*Example:*

*Create keyspace*Student *with replication*={*'class*':’SimpleStrategy’, '*replication\_factor*': 1};

* **Selecting Keyspace for Cassandra Table**

*Syntax:*

USE *keyspace\_name*;

or

Specify the Keyspace Name in the Query

CREATE TABLE *keyspace\_name*.*table\_name*

*Example:*

CREATE TABLE Super\_Keyspace.student;

1. **Commands using In Built Data Type**

**How to Create Cassandra Table**

*Syntax:*

CREATE TABLE tableName (

columnName1dataType,

columnName2dataType,

columnName2datatype PRIMARY KEY (columnName) );

*example:*

CREATE TABLE suppliers (

supp\_idint PRIMARY KEY,

supp\_city text,

supp\_email text,

supp\_feeint,

supp\_name text,

supp\_phoneint );

**Updating Data in a Table**

UPDATE <tablename> SET <column name> = <new value><column name> = <value>.... WHERE <condition>

**Deleting Datafrom a Table**

DELETE FROM <identifier> WHERE <condition>;

Deleting an Entire Row

DELETE FROM emp WHERE emp\_id=3;

**1. Commands using List Data Type**

**Creating a Table with List**

CREATE TABLE data(name text PRIMARY KEY, email list<text>);

**Inserting Data into a List**

INSERT INTO data(name, email) VALUES ('ramu',['abc@gmail.com','cba@yahoo.com'])

**Updating a List**

UPDATE data SET email = email + ['xyz@tutorialspoint.com'] where name = 'ramu'

1. **Commands using Set Data Type**

**Creating a Table with Set**

CREATE TABLE data2 (name text PRIMARY KEY, marks set<varint>);

**Inserting Data into a Set**

INSERT INTO data2(name, marks)VALUES (‘roshan', {38,39});

**Updating a Set**

UPDATE data2 SET marks = marks + {33} where name = 'roshan’;

1. **Commands using Map Data Type**

**Creating a Table with Map**

CREATE TABLE data3 (name text PRIMARY KEY, address map<timestamp, text>);

**Inserting Data into a Map**

INSERT INTO data3 (name, address) VALUES ('robin', {'home' : 'hyderabad' , 'office' : 'Delhi' } );

**Updating a Map**

UPDATE data3 SET address = address + {'office':'mumbai'} WHERE name = 'robin';

**CONCLUSIONS:**

By following above Steps students will able to install Cassandra on Windows. Students will be able to perform basic operations using data types and commands of Cassandra.

**Additional Task :**

1. **Install cassandra on Linux operating system**
2. **Design user interface for cassandra using either Java or Python.**
3. **Compare and contrast NoSQL with SQL**

**Journal Write-up**

* **Introduction**
* **SQL vs NoSQL**
* **History and Terminology of Cassandra**
* **Cassandra Basic commands**
* **Design a database schema for any application of your choice**
* **Solve minimum 30 queries which includes all aspect of Cassandra**
* **Cassandra connectivity with Java & Python**
* **Conclusion**

**3. Lab Exercise**

Exercise No 3: (2 Hours) – 1 Practical

# Aim: - Cloudera setup and Installation using VM

# Objectives:

1. Students should able to install Cloudera using VM

2. Students should able to use HDFS file system using Cloudera

3. Student should able to use Map-Reduce paradigm from Cloudera

**THEORY:**

**Cloudera, Inc.** is a US-based software company that provides a software platform for data engineering, data warehousing, machine learning and analytics that runs in the cloud or on premises.

Cloudera started as a hybrid open-source Apache Hadoop distribution, CDH (Cloudera Distribution Including Apache Hadoop), that targeted enterprise-class deployments of that technology. Cloudera states that more than 50% of its engineering output is donated upstream to the various Apache-licensed open source projects (Apache Spark, Apache Hive, Apache Avro, Apache HBase, and so on) that combine to form the Apache Hadoop platform. Cloudera is also a sponsor of the Apache Software Foundation. Following diagram shows taxonomy of Cloudera Distribution Including Apache Hadoop.

 Cloudera solution taxonomy
CLOUDERA HADOOP ARCHITECTURE
 

**Installation steps to be followed: (Windows)**

You can install CDH4 in any of the following ways:

1. Installing using Cloudera quickstart vm.
2. Automated method using Cloudera Manager. Cloudera Manager Free Edition automates the installation and configuration of CDH4 on an entire cluster if you have root or password less sudo SSH access to your cluster's machines.
3. Manual methods described below: -
   1. Download and install the CDH4 "1-click Install" package
   2. Add the CDH4 repository
   3. Build your own CDH4 repository
   4. Install from a CDH4 tarball

**Downloading and Installing the Cloudera VM Instructions**

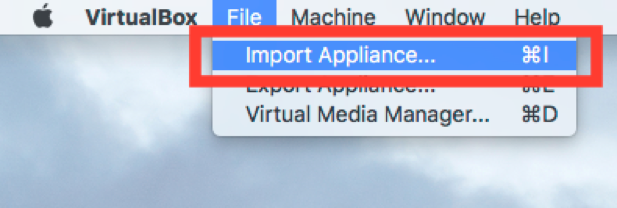
**Hardware Requirements:**

1. Quad Core Processor (VT-x or AMD-V support recommended), 64-bit;
2. 8 GB RAM
3. 20 GB disk free.
4. A high speed internet connection because you will be downloading files up to 4 Gb in size.

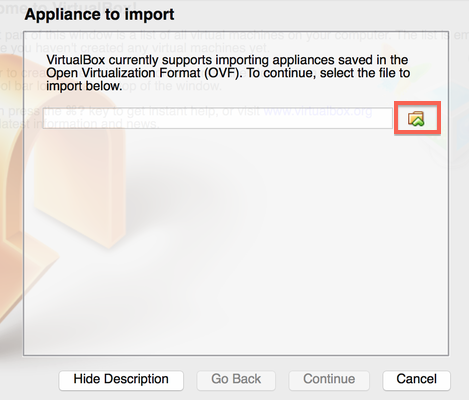
**Instructions**

Please use the following instructions to download and install the Cloudera Quickstart VM with VirutalBox.. The screenshots are from a Mac but the instructions should be the same for Windows.

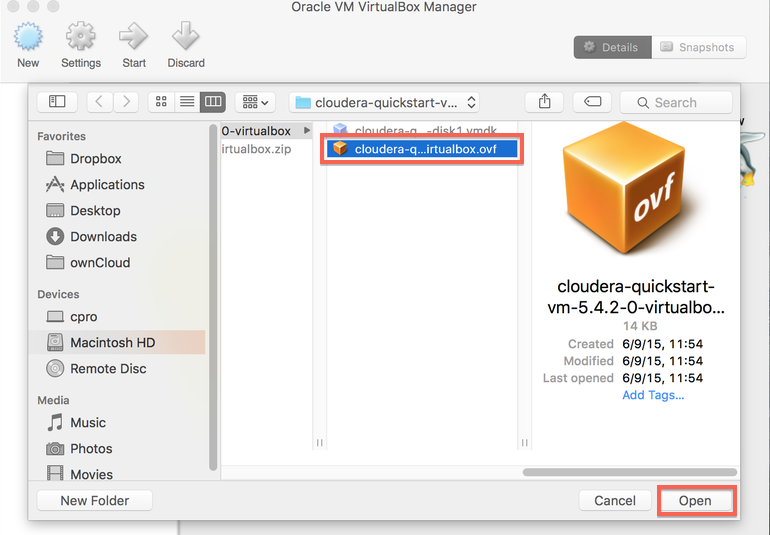
1. **Install VirtualBox.**
   1. Go to <https://www.virtualbox.org/wiki/Downloads> to download and install VirtualBox for your computer.
2. **Download the Cloudera VM.**
   1. Download the Cloudera VM from  [https://downloads.cloudera.com/demo\_vm/virtualbox/cloudera-quickstart-vm-5.4.2-0-virtualbox.zip](https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.4.2-0-virtualbox.zip" \t "_blank). The VM is over 4GB, so will take some time to download.
3. **Unzip the Cloudera VM**:
   1. On Windows: Right-click cloudera-quickstart-vm-5.4.2-0-virtualbox.zip and select “Extract All…”
4. **Start VirtualBox**.
5. **Begin importing**.
   1. Import the VM by going to File -> Import Appliance



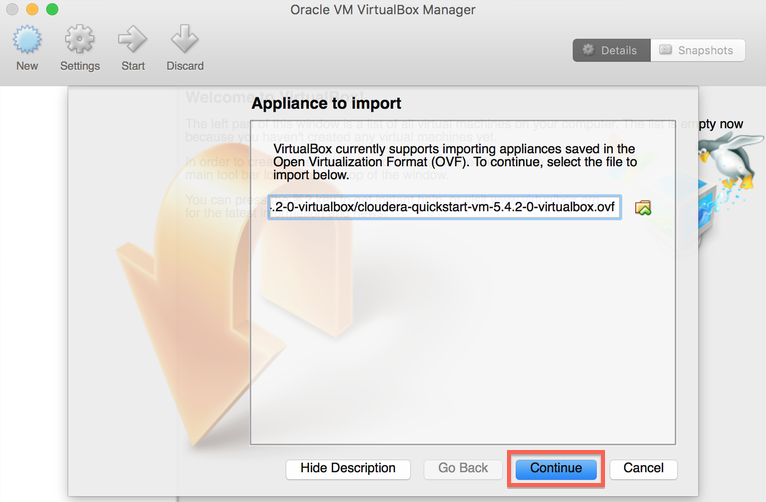
6. **Click the Folder icon.**



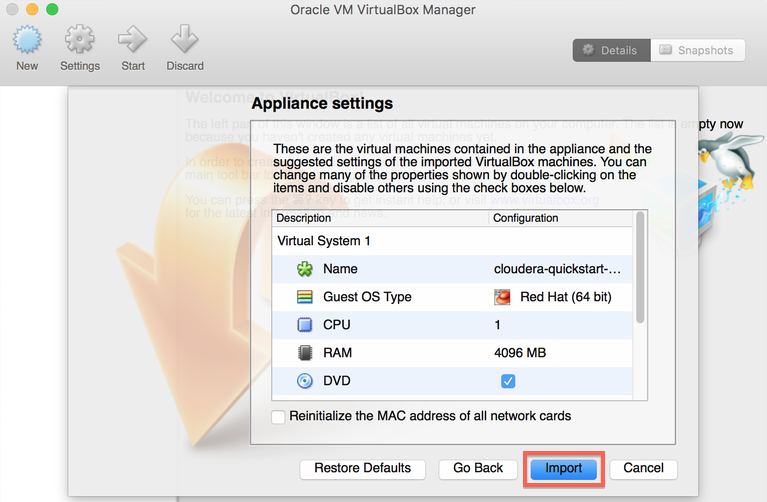
7. **Select the cloudera-quickstart-vm-5.4.2-0-virtualbox.ovf** from the Folder where you unzipped the VirtualBox VM and click Open.



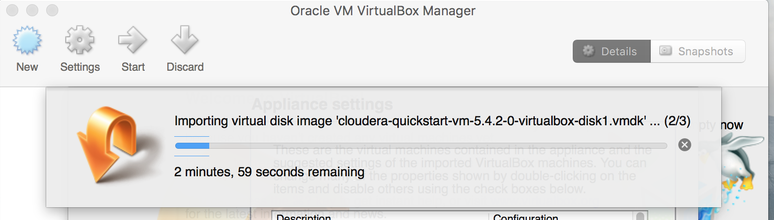
8. **Click Continue** to proceed.



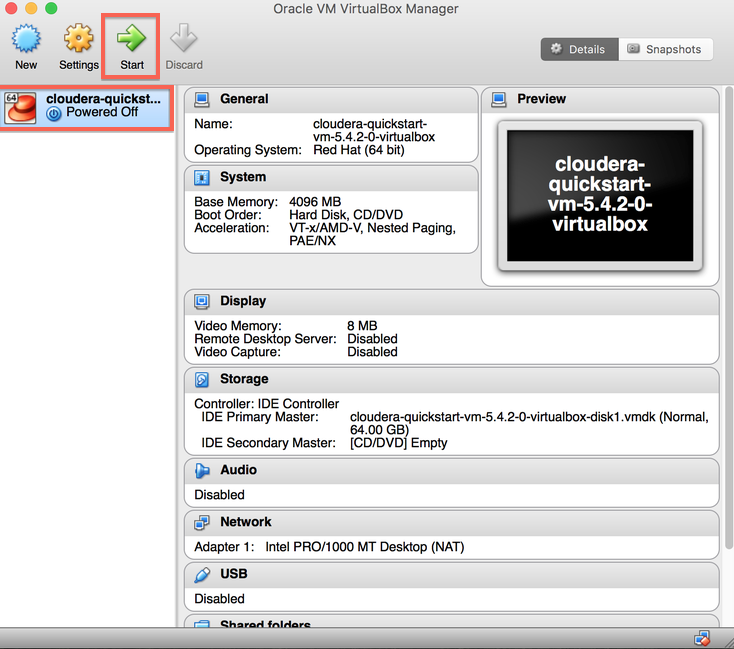
9. **Click Import.**



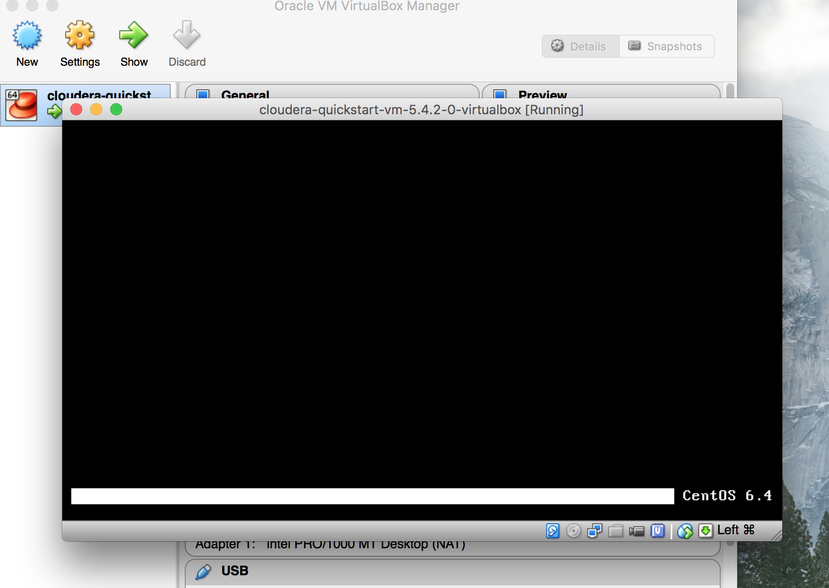
10. **The virtual machine image will be imported.** This can take several minutes.



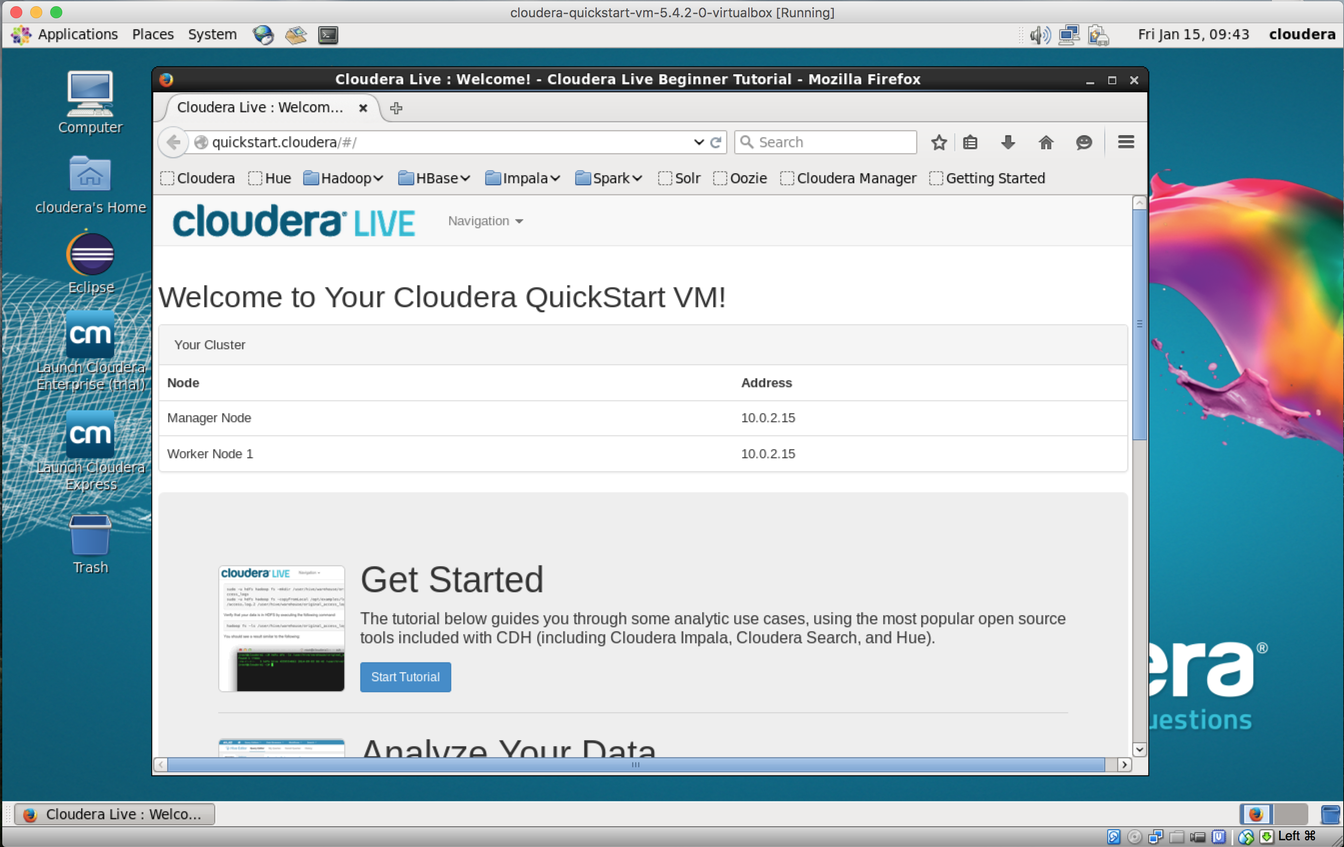
11. **Launch Cloudera VM.** When the importing is finished, the quickstart-vm-5.4.2-0 VM will appear on the left in the VirtualBox window. Select it and click the Start button to launch the VM.



12. **Cloudera VM booting.**It will take several minutes for the Virtual Machine to start. The booting process takes a long time since many Hadoop tools are started.



13. **The Cloudera VM desktop.**Once the booting process is complete, the desktop will appear with a browser.



**CONCLUSIONS:**

In this Practical we learned how to install and enable Cloudera Distribution Including Apache Hadoop using VM.

**Journal Write-up**

* **Introduction**
* **Cloudera Hadoop Architecture**
* **Installation of cloudera on Windows**
* **User Interface in cloudera**
* **Map Reduce using cloudera**
* **Using Apache Hive in cloudera**
* **Using Apache Pig in cloudera**
* **Conclusion**

**4. Lab Exercise**

Exercise No 4: (2 Hours) – 1 Practical

**Aim: - Setup and installation of Apache Hadoop on Windows**.

**THEORY:**

The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is a 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 single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

# Prerequisites :

1. Java 8 runtime environment (JRE): [Hadoop 3 requires a Java 8 installation](https://cwiki.apache.org/confluence/display/HADOOP/Hadoop+Java+Versions).

2. [Java 8 development Kit (JDK)](https://www.oracle.com/java/technologies/javase-jdk8-downloads.html)

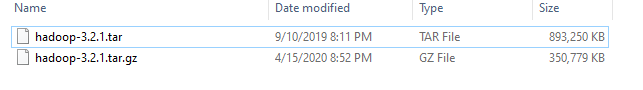
3. To unzip downloaded Hadoop binaries,

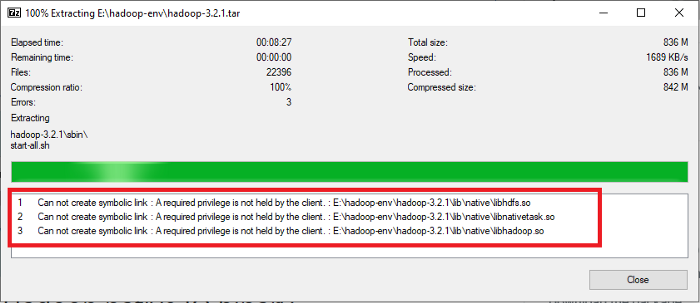
4. Create a folder “D:\hadoopSoft” on local machine to store downloaded files.

# Download Hadoop binaries

The first step is to download Hadoop binaries from the [official website](https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz). The binary package size is about 342 MB. After finishing the file download, unpack the package in two steps. First, extract the hadoop-3.2.1.tar.gz library, and then, unpack the extracted tar file

The tar file extraction may take some minutes to finish. In the end, you may see some warnings about symbolic link creation. Just ignore these warnings since they are not related to windows.





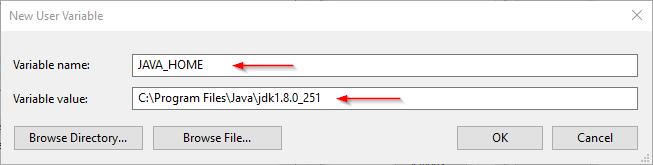
# Setting up environment variables

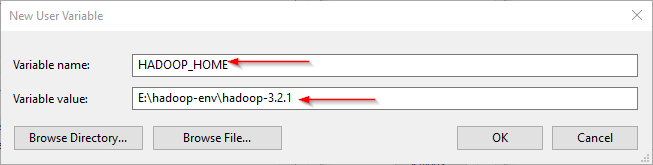
After installing Hadoop and its prerequisites, we should configure the environment variables to define Hadoop and Java default paths.

There are two variables to define:

1. JAVA\_HOME: JDK installation folder path

2. HADOOP\_HOME: Hadoop installation folder path





# Configuring Hadoop cluster

There are four files we should alter to configure Hadoop cluster:

1. %HADOOP\_HOME%\etc\hadoop\hdfs-site.xml
2. %HADOOP\_HOME%\etc\hadoop\core-site.xml
3. %HADOOP\_HOME%\etc\hadoop\mapred-site.xml
4. %HADOOP\_HOME%\etc\hadoop\yarn-site.xml

**HDFS site configuration**

Hadoop is built using a master-slave paradigm. Before altering the HDFS configuration file, create a directory to store all master node (name node) data and another one to store data (data node). In this example, created the following directories:

* D\hadoop-soft\hadoop-3.2.1\data\dfs\namenode
* D:\hadoop-soft\hadoop-3.2.1\data\dfs\datanode

Open “hdfs-site.xml” file located in “%HADOOP\_HOME%\etc\hadoop” directory, and add the following properties within the <configuration></configuration> element:

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:///D:/hadoop-soft/hadoop-3.2.1/data/dfs/namenode</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:///D:/hadoop-soft/hadoop-3.2.1/data/dfs/datanode</value>

</property>

**Core site configuration**

Now, configure the name node URL adding the following XML code into the <configuration></configuration> element within “core-site.xml”:

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9820</value>

</property>

**Map Reduce site configuration**

Now,add the following XML code into the <configuration></configuration> element within “mapred-site.xml”:

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

<description>MapReduce framework name</description>

</property>

**Yarn site configuration**

Now, add the following XML code into the <configuration></configuration> element within “yarn-site.xml”:

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

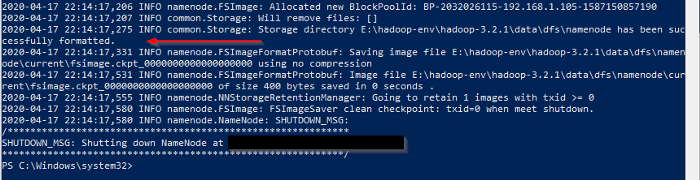
<description>Yarn Node Manager Aux Service</description>

</property>

# Formatting Name node

After finishing the configuration, format the name node using the following command:

*hdfs namenode –format*

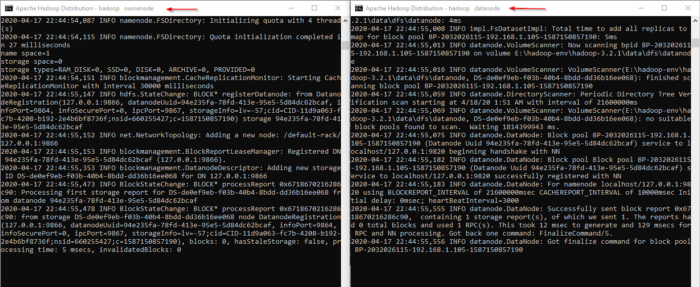


# Starting Hadoop services

Now, open command prompt , and navigate to “%HADOOP\_HOME%\sbin” directory. Then the following command to start the Hadoop nodes:

***.\start-dfs.cmd***

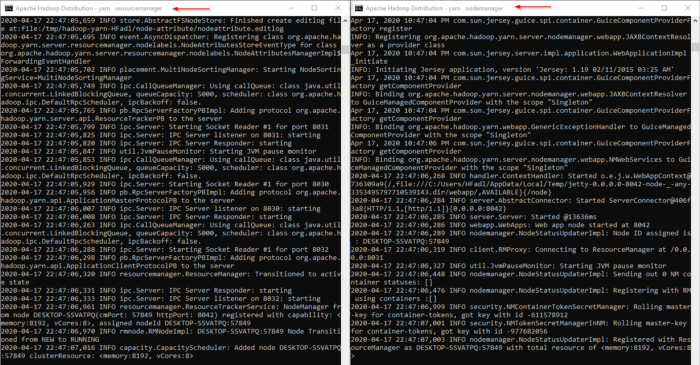
Two command prompt windows will open (one for the name node and one for the data node) as follows:



Next, start the Hadoop Yarn service using the following command:

***./start-yarn.cmd***

Two command prompt windows will open (one for the resource manager and one for the node manager) as follows:



To make sure that all services started successfully, we can run the following command:

***jps***

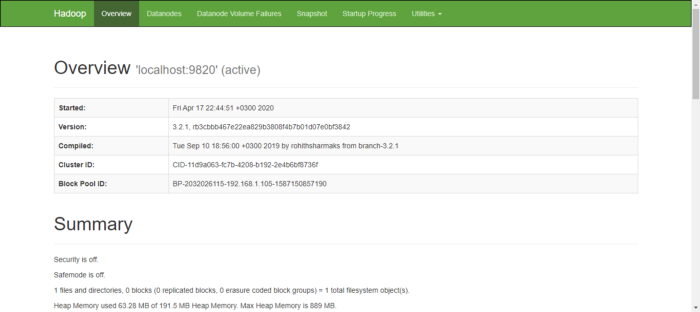
It should display the following services:

14560 DataNode  
4960 ResourceManager  
5936 NameNode  
768 NodeManager  
14636 Jps

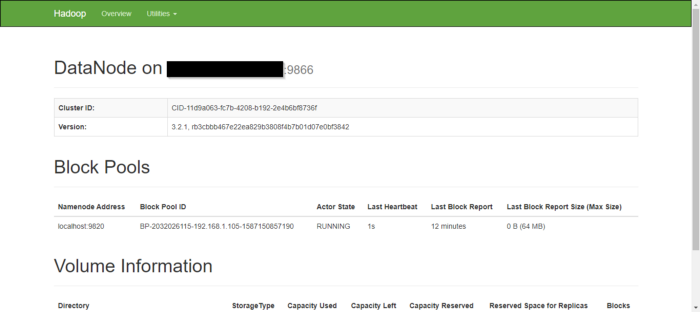
# Hadoop Web UI

There are three web user interfaces to be used:

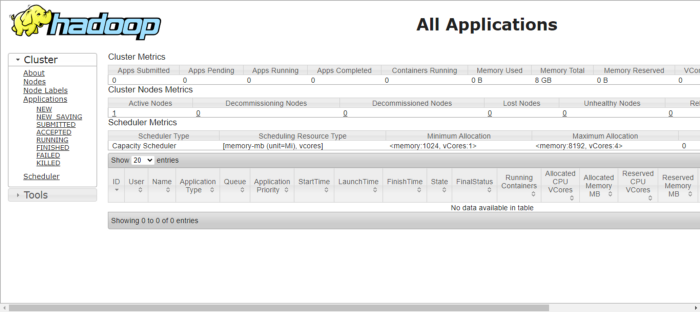
* Name node web page: <http://localhost:9870/dfshealth.html>



Data node web page: <http://localhost:9864/datanode.html>



Yarn web page: <http://localhost:8088/cluster>



**CONCLUSIONS:**

Student will able to install Hadoop framework by using above steps.

**Journal Write-up**

* **Introduction**
* **History of Hadoop**
* **Hadoop Ecosysytem**
* **HDFS Architecture**
* **Functions of Name Node and Data Node**
* **Hadoop Installation Process**
* **HDFS Command Inteface**
* **Conclusion**

**5. Lab Exercise**

Exercise No 4: (2 Hours) – 1 Practical

1. **Aim: - File management task in Apache Hadoop**

**THEORY:**

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel with others. In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.

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).

v

**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 Distributed File System**

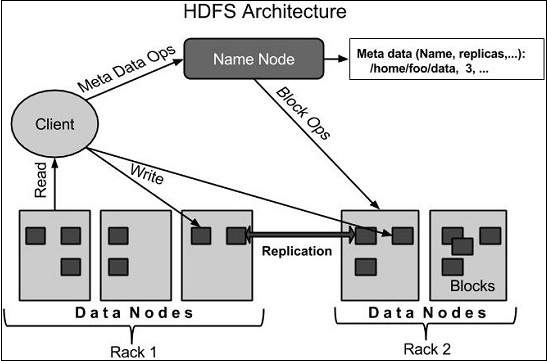
The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. 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.

**HDFS Architecture**

Given below is the architecture of a Hadoop File System.



**Starting HDFS**

***.\start-dfs.cmd***

**Listing Files in HDFS**

After loading the information in the server, 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>***

|  |  |
| --- | --- |
| **Sr.No** | **Command & Description** |
| 1 | **-ls <path>**  Lists the contents of the directory specified by path, showing the names, permissions, owner, size and modification date for each entry. |
| 2 | **-lsr <path>**  Behaves like -ls, but recursively displays entries in all subdirectories of path. |
| 3 | **-du <path>**  Shows disk usage, in bytes, for all the files which match path; filenames are reported with the full HDFS protocol prefix. |
| 4 | **-dus <path>**  Like -du, but prints a summary of disk usage of all files/directories in the path. |
| 5 | **-mv <src><dest>**  Moves the file or directory indicated by src to dest, within HDFS. |
| 6 | **-cp <src> <dest>**  Copies the file or directory identified by src to dest, within HDFS. |
| 7 | **-rm <path>**  Removes the file or empty directory identified by path. |
| 8 | **-rmr <path>**  Removes the file or directory identified by path. Recursively deletes any child entries (i.e., files or subdirectories of path). |
| 9 | **-put <localSrc> <dest>**  Copies the file or directory from the local file system identified by localSrc to dest within the DFS. |
| 10 | **-copyFromLocal <localSrc> <dest>**  Identical to –put |
| 11 | **-moveFromLocal <localSrc> <dest>**  Copies the file or directory from the local file system identified by localSrc to dest within HDFS, and then deletes the local copy on success. |

**CONCLUSIONS:**

Students will able to perform file handling commands on the Hadoop platform.

**Journal Write-up**

* **Introduction**
* **Role of HDFS and YARN in Hadoop**
* **HDFS Commands**
* **Administrative Commands**
* **Conclusion**

**6. Lab Exercise**

Exercise No 4: (2 Hours) – 1 Practical

1. **Aim: - Map Reduce Paradigm**

.**THEORY:**

MapReduce is a framework used to write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner.

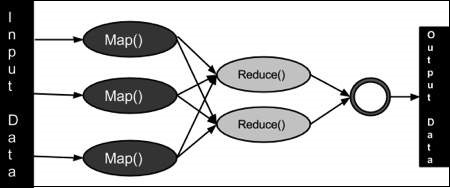
**What is MapReduce?**

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

**The Algorithm**

* Generally MapReduce paradigm is based on sending the computer to where the data resides!
* MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
  + **Map stage** − The map or mapper’s job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.
  + **Reduce stage** − This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer’s job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.
* During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
* The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
* Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
* After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.



**CONCLUSIONS:**

Students able to write simple programs using map-reduce paradigm.

**Journal Write-up**

* **Introduction**
* **History of Map Reduce**
* **Map Reduce jar file in Hadoop**
* **Map Reduce using Java**
* **Map Reduce using Python**
* **Conclusion**

**7. Lab Exercise**

Exercise No 4: (2 Hours) – 1 Practical

**Aim: - Install, Deploy and configure Apache Spark. Develop application using Apache Spark.**

**THEORY:**

Apache Spark is an open-source framework that processes large volumes of stream data from multiple sources. Spark is used in distributed computing with machine learning applications, data analytics, and graph-parallel processing.

Apache Spark is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including [Spark SQL](https://spark.apache.org/docs/latest/sql-programming-guide.html) for SQL and structured data processing, [MLlib](https://spark.apache.org/docs/latest/ml-guide.html) for machine learning, [GraphX](https://spark.apache.org/docs/latest/graphx-programming-guide.html) for graph processing, and [Spark Streaming](https://spark.apache.org/docs/latest/streaming-programming-guide.html).

**Installation of Apache Spark:**

**Prerequisites**

* A system running Windows 10
* A user account with administrator privileges (required to install software, modify file permissions, and modify system PATH)
* Command Prompt
* A tool to extract .tar files
* Java 8
* **Python 3.8**

### Download Apache Spark

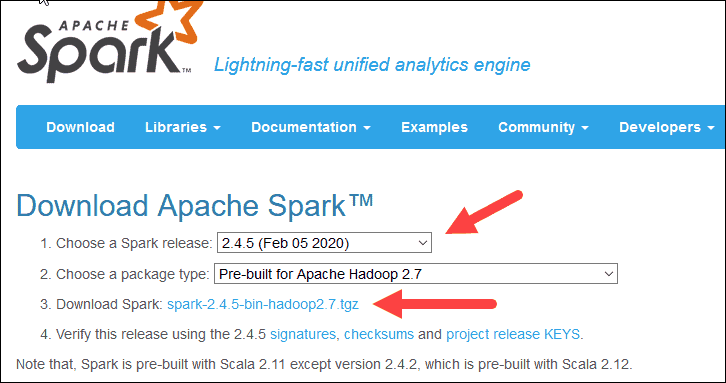
1. Open a browser and navigate to <https://spark.apache.org/downloads.html>.

2. Under the Download Apache Spark heading, there are two drop-down menus. Use the current non-preview version.

* In our case, in ***Choose a Spark release*** drop-down menu select **2.4.5 (Feb 05 2020)**.
* In the second drop-down **Choose a package type,** leave the selection **Pre-built for Apache Hadoop 2.7**.

3. Click the **spark-2.4.5-bin-hadoop2.7.tgz**link.

4. A page with a list of mirrors loads where you can see different servers to download from. Pick any from the list and save the file to your Downloads folder.



### Install Apache Spark

Installing Apache Spark involves **extracting the downloaded file** to the desired location.

1. Create a new folder named Spark in the root of your C: drive.

2. In Explorer, locate the Spark file you downloaded.

3. Right-click the file and extract it to C:\Spark using the tool you have on your system

4. Now, your C:\Spark folder has a new folder spark-2.4.5-bin-hadoop2.7 with the necessary files inside.

### Add winutils.exe File

Download the **winutils.exe** file for the underlying Hadoop version for the Spark installation you downloaded.

1. Navigate to this URL <https://github.com/cdarlint/winutils> and inside the **bin** folder, locate **winutils.exe**, and click it.

2. Find the **Download** button on the right side to download the file.

3. Now, create new folders **Hadoop** and **bin** on C: using Windows Explorer or the Command Prompt.

4. Copy the winutils.exe file from the Downloads folder to **C:\hadoop\bin**.



### Configure Environment Variables

**Set the *SPARK\_HOME***, ***HADOOP\_HOME & JAVA\_HOME***  to appropriate folders.

### Launch Spark

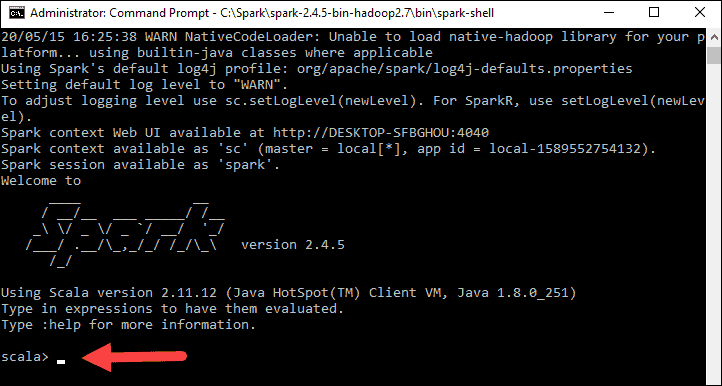
1. Open a new command-prompt window

2. To start Spark, enter:

***C:\Spark\spark-2.4.5-bin-hadoop2.7\bin\spark-shell***

3. The system should display several lines indicating the status of the application. You may get a Java pop-up. Select **Allow access** to continue.

Finally, the Spark logo appears, and the prompt displays the **Scala shell**.



**CONCLUSIONS:**

Students will able to install, configure and use Apache Spark for simple tasks.

**Journal Write-up**

* **Introduction**
* **Spark Streaming**
* **History and Installation of Apache Spark**
* **Spark Streaming Fundamentals**
* **Application Development using Apache Spark**
* **Conclusion**

**8. Lab Exercise**

Exercise No 4: (2 Hours) – 1 Practical

**Aim: - Application development using Apache PySpark**.

**THEORY:**

Apache Spark is a new and open-source framework used in the big data industry for real-time processing and batch processing. It supports different languages, like Python, Scala, Java, and R.

Apache Spark is initially written in a Java Virtual Machine (JVM) language called Scala, whereas Pyspark is like a Python API which contains a library called Py4J. This allows dynamic interaction with JVM objects.

**Pyspark = Python + Apache Spark**

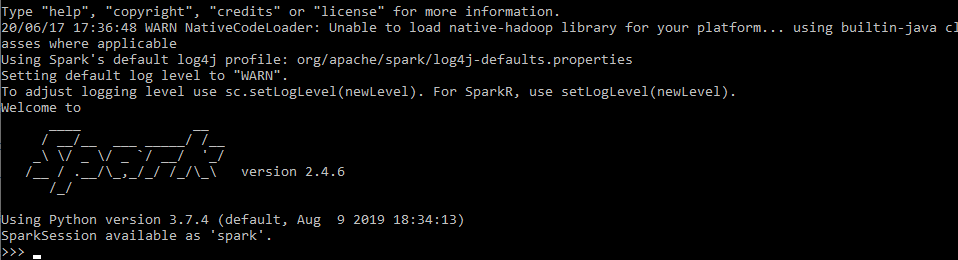
**Installation of PySpark is similar to that of Apache Spark as explain in 7’Th experiment.**

**Starting of PySpark**

Open Command Prompt and type the following command

***Pyspark***

Following message is displayed and pyspark is ready to execute task.



### Who uses PySpark?

PySpark is very well used in Data Science and Machine Learning community as there are many widely used data science libraries written in Python including NumPy, TensorFlow also used due to its efficient processing of large datasets. PySpark has been used by many organizations like Walmart, Trivago, Sanofi, Runtastic, and many more.

**PySpark Modules & Packages**

* PySpark RDD ([pyspark.RDD](https://spark.apache.org/docs/latest/api/python/pyspark.html#pyspark.RDD))
* PySpark DataFrame and SQL ([pyspark.sql](https://spark.apache.org/docs/latest/api/python/pyspark.sql.html))
* PySpark Streaming ([pyspark.streaming](https://spark.apache.org/docs/latest/api/python/pyspark.streaming.html))
* PySpark MLib ([pyspark.ml](https://spark.apache.org/docs/latest/api/python/pyspark.ml.html), [pyspark.mllib](https://spark.apache.org/docs/latest/api/python/pyspark.mllib.html))
* PySpark GraphFrames ([GraphFrames](https://graphframes.github.io/graphframes/docs/_site/index.html))
* PySpark Resource ([pyspark.resource](https://spark.apache.org/docs/latest/api/python/pyspark.resource.html)) It’s new in PySpark 3.0

**PySpark RDD – Resilient Distributed Dataset**

PySpark RDD (Resilient Distributed Dataset) is a fundamental data structure of PySpark that is fault-tolerant, immutable distributed collections of objects, which means once you create an RDD you cannot change it. Each dataset in RDD is divided into logical partitions, which can be computed on different nodes of the cluster.

### RDD Creation

In order to create an RDD, first, you need to create a SparkSession which is an entry point to the PySpark application. SparkSession can be created using a builder() or newSession() methods of the SparkSession.

***spark = SparkSession.builder()***

***.master("local[1]")***

***.appName("SparkByExamples.com")***

***.getOrCreate()***

#### using parallelize()

SparkContext has several functions to use with RDDs. For example, it’s ***parallelize()*** method is used to create an RDD from a list.

***#Create RDD from parallelize***

***dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]***

***rdd=spark.sparkContext.parallelize(dataList)***

#### Using textFile()

RDD can also be created from a text file using ***textFile()*** function of the SparkContext.

***//Create RDD from external Data source***

***rdd2 = spark.sparkContext.textFile("/path/textFile.txt")***

### RDD Operations

On PySpark RDD, following operations are perform.

**RDD transformations –** Transformations are lazy operations. When you run a transformation(for example update), instead of updating a current RDD, these operations return another RDD.

**RDD actions** – operations that trigger computation and return RDD values to the driver.

#### RDD Transformations

Transformations on Spark RDD returns another RDD and transformations are lazy meaning they don’t execute until you call an action on RDD. Some transformations on RDD’s are flatMap(), map(), reduceByKey(), filter(), sortByKey() and return new RDD instead of updating the current.

#### RDD Actions

RDD Action operation returns the values from an RDD to a driver node. In other words, any RDD function that returns non RDD[T] is considered as an action.

Some actions on RDD’s are count(), collect(), first(), max(), reduce() and more.

**CONCLUSIONS:**

Student will able to install Pyspark, create RDD using different techniques and perform relevant operations on it.

**Journal Write-up**

* **Introduction**
* **What is Apache PySpark**
* **PySpark Framework**
* **PySpark SQL**
* **Transformation and Actions**
* **Data Frame**
* **Conclusion**