|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | | |
|  | Name of the student: | | Yash Sankpal | | | Roll No. | | | 8695 | |  |
| Practical Number: 1 | |  | | | Date of Practical: | | |  | |
| Relevant CO’s:  ITC802.2 | | At the end of the course students will be able to use tools like hadoop and NoSQL to solve big data related problems. | | | | | | | |
| Sign here to indicate that you have read all the relevant material provided Sign:  before attempting this practical Practical grading using Rubrics  Indicator Very Poor Poor Average Good Excellent | | | | | | | | | |
| Timeline (2) | More than a  session late  (0) | | NA (0.5) | NA(1) | | NA (1.5) | Early or on  time (2) | |  |
| Completeness  (3) | N/A | | N/A | Not Com-  pleted (1) | | Partially  Completed  (2) | Completed(3) | |
| Legibility (3) | N/A | | N/A | poor(1) | | Good(2) | Very Good  (3) | |
| Postlab (2) | N/A | | N/A | N/A | | Partially  Correct(1) | All correct answers (2) | |
| |  |  | | --- | --- | | Total Marks (10) | Sign of instructor | |  |  | | | | | | | | | |

Practical

Course title: Big Data Analytics

Course term: 2021-2022

Problem Statement: Study of hadoop installation, eclipse plugin configuration for hadoop and hadoop ecosystem.

Theory:Write about components of hadoop

**HDFS:** 

* HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.
* HDFS consists of two core components i.e.
  1. Name node
  2. Data Node
* Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores 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.

**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.
  1. Resource Manager
  2. Nodes Manager
  3. 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.

**MapReduce:** 

* By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing’s logic and helps to write applications which transform big data sets into a manageable one.
* MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:
  1. *Map()* performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
  2. *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 smaller set of tuples.

**PIG:**

 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. Just the way Java runs on the [JVM](https://www.geeksforgeeks.org/jvm-works-jvm-architecture/).
* Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

**HIVE:** 

* With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
* It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
* Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
* JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

**Mahout:** 

* Mahout, allows Machine Learnability to a system or application. [Machine Learning](https://write.geeksforgeeks.org/geek/ml-what-is-machine-learning-2/), as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or on the basis of algorithms.
* It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

**Apache Spark:** 

* 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.
* It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
* Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably.

**Apache HBase:** 

* 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

**Other Components:** Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:

* **Solr, Lucene:**These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.
* **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, inter-component based communication, grouping, and maintenance.
* **Oozie:**Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.

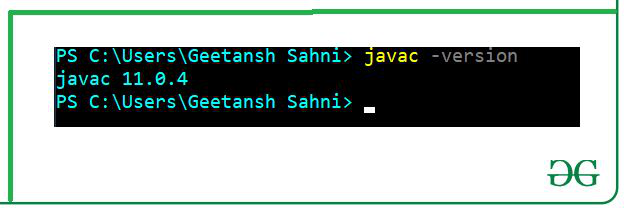
Explain the step by step procedure of hadoop installation.

**Prerequisite:**

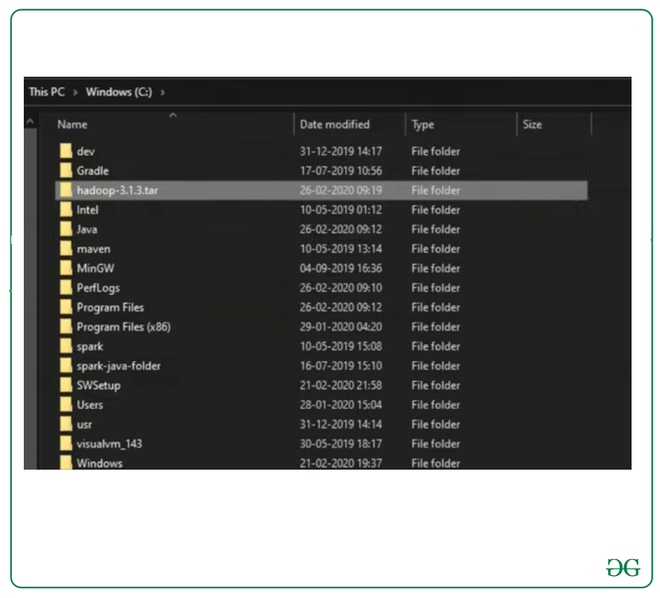
1. JAVA-Java JDK ([installed](https://www.geeksforgeeks.org/setting-environment-java/))
2. HADOOP-Hadoop package ([Downloaded](https://hadoop.apache.org/releases.html))

**Step 1: Verify the Java installed**

javac -version

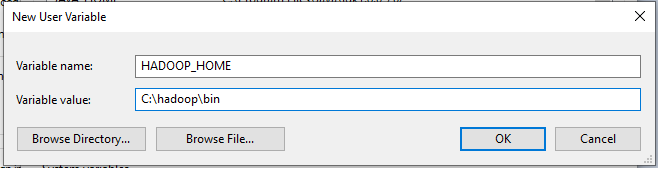


**Step 2: Extract Hadoop at C:\Hadoop**



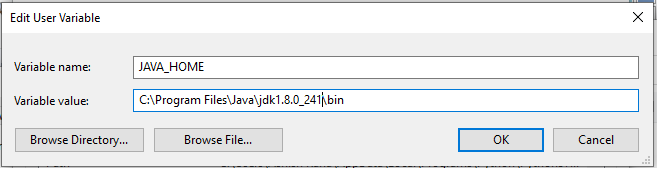
**Step 3:** **Setting up the HADOOP\_HOME variable**

Use windows environment variable setting for Hadoop Path setting.

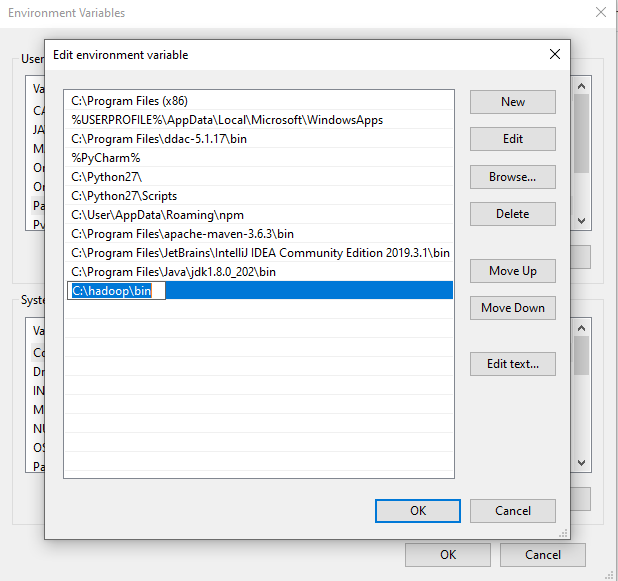


**Step 4: Set JAVA\_HOME variable**

Use windows environment variable setting for Hadoop Path setting.



**Step 5:** **Set Hadoop and Java bin directory path**



**Step 6: Hadoop Configuration :**

For Hadoop Configuration we need to modify Six files that are listed below-

1. Core-site.xml

2. Mapred-site.xml

3. Hdfs-site.xml

4. Yarn-site.xml

5. Hadoop-env.cmd

6. Create two folders datanode and namenode

**Step 6.1: Core-site.xml configuration**

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

**Step 6.2: Mapred-site.xml configuration**

<configuration>

<property>

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

<value>yarn</value>

</property>

</configuration>

**Step 6.3: Hdfs-site.xml configuration**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

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

<value>C:\hadoop-2.8.0\data\namenode</value>

</property>

<property>

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

<value>C:\hadoop-2.8.0\data\datanode</value>

</property>

</configuration>

**Step 6.4: Yarn-site.xml configuration**

<configuration>

<property>

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

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>

**Step 6.5: Hadoop-env.cmd configuration**

Set "JAVA\_HOME=C:\Java" (On C:\java this is path to file jdk.18.0)

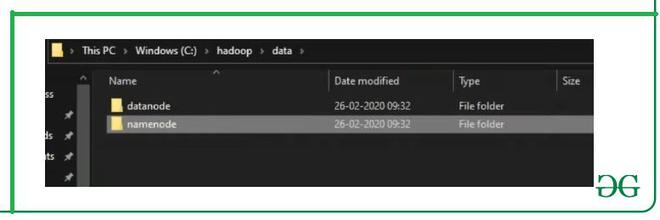
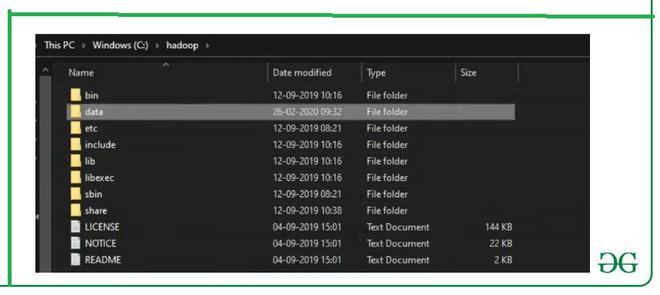


**Step 6.6: Create datanode and namenode folders**

1. Create folder "data" under "C:\Hadoop-2.8.0"

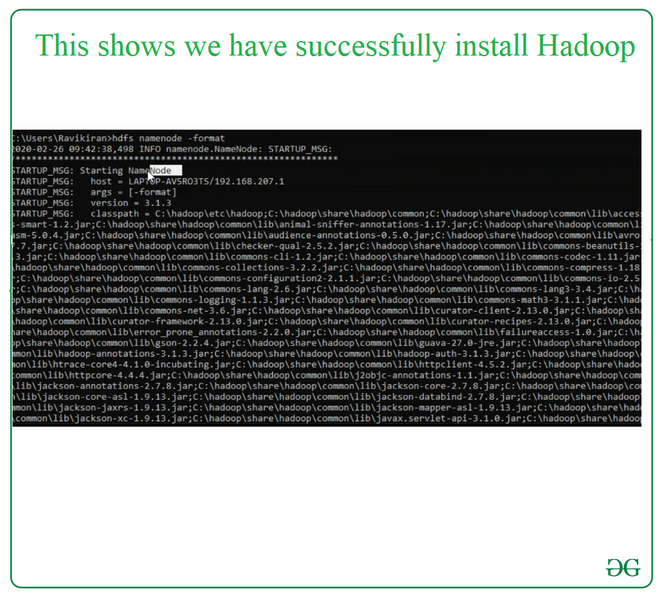
2. Create folder "datanode" under "C:\Hadoop-2.8.0\data"

3. Create folder "namenode" under "C:\Hadoop-2.8.0\data"



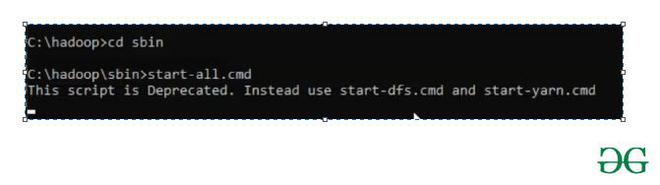
**Step 7: Format the namenode folder**

Open command window (cmd) and typing command “hdfs namenode –format”



**Step 8: Testing the setup**

Open command window (cmd) and typing command “start-all.cmd”



Explain the step by step procedure of installing and configuring eclipse plugin for hadoop.

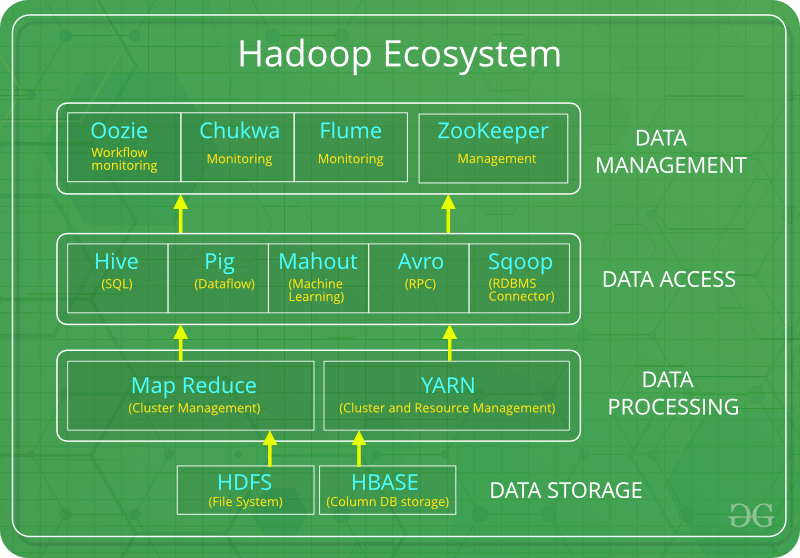
* Download eclipse from [**http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops4/R-4.10-201812060815/eclipse-SDK-4.10-win32-x86\_64.zip**](http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops4/R-4.10-201812060815/eclipse-SDK-4.10-win32-x86_64.zip)
* Extract the zip file and click on eclipse .exe file.
* Download .jar file of eclipse Hadoop plugin and copy paste it at plugins folder of sub folder of eclipse file.
* Then restart eclipse.
* After restarting click on open perspective tool and add map/reduce.
* After opening create hdfs local file and start Hadoop system.
* Connection with Hadoop with eclipse will show us it is perfectly configured.

PostLab:Explain hadoop ecosystem in detail.

*Hadoop Ecosystem*is a platform or a suite which provides various services to solve the 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.

Following are the components that collectively form a Hadoop ecosystem: 

* **HDFS:**Hadoop Distributed File System
* **YARN:** Yet Another Resource Negotiator
* **MapReduce:** Programming based Data Processing
* **Spark:** In-Memory data processing
* **PIG, HIVE:** Query based processing of data services
* **HBase:**NoSQL Database
* **Mahout, Spark MLLib:** [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)algorithm libraries
* **Solar, Lucene:** Searching and Indexing
* **Zookeeper:** Managing cluster
* **Oozie:** Job Scheduling



**Note:** Apart from the above-mentioned components, there are many other components too that are part of the Hadoop ecosystem.

All these toolkits or components revolve around one term i.e. *Data*. That’s the beauty of Hadoop that it revolves around data and hence making its synthesis easier.

**HDFS:** 

* HDFS is the primary or major component of Hadoop ecosystem and is responsible for storing large data sets of structured or unstructured data across various nodes and thereby maintaining the metadata in the form of log files.
* HDFS consists of two core components i.e.
  1. Name node
  2. Data Node
* Name Node is the prime node which contains metadata (data about data) requiring comparatively fewer resources than the data nodes that stores 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.

**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.
  1. Resource Manager
  2. Nodes Manager
  3. 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.

**MapReduce:** 

* By making the use of distributed and parallel algorithms, MapReduce makes it possible to carry over the processing’s logic and helps to write applications which transform big data sets into a manageable one.
* MapReduce makes the use of two functions i.e. Map() and Reduce() whose task is:
  1. *Map()* performs sorting and filtering of data and thereby organizing them in the form of group. Map generates a key-value pair based result which is later on processed by the Reduce() method.
  2. *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 smaller set of tuples.

**PIG:**

 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. Just the way Java runs on the [JVM](https://www.geeksforgeeks.org/jvm-works-jvm-architecture/).
* Pig helps to achieve ease of programming and optimization and hence is a major segment of the Hadoop Ecosystem.

**HIVE:** 

* With the help of SQL methodology and interface, HIVE performs reading and writing of large data sets. However, its query language is called as HQL (Hive Query Language).
* It is highly scalable as it allows real-time processing and batch processing both. Also, all the SQL datatypes are supported by Hive thus, making the query processing easier.
* Similar to the Query Processing frameworks, HIVE too comes with two components: *JDBC Drivers* and *HIVE Command Line*.
* JDBC, along with ODBC drivers work on establishing the data storage permissions and connection whereas HIVE Command line helps in the processing of queries.

**Mahout:** 

* Mahout, allows Machine Learnability to a system or application. [Machine Learning](https://write.geeksforgeeks.org/geek/ml-what-is-machine-learning-2/), as the name suggests helps the system to develop itself based on some patterns, user/environmental interaction or on the basis of algorithms.
* It provides various libraries or functionalities such as collaborative filtering, clustering, and classification which are nothing but concepts of Machine learning. It allows invoking algorithms as per our need with the help of its own libraries.

**Apache Spark:** 

* 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.
* It consumes in memory resources hence, thus being faster than the prior in terms of optimization.
* Spark is best suited for real-time data whereas Hadoop is best suited for structured data or batch processing, hence both are used in most of the companies interchangeably.

**Apache HBase:** 

* 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

**Other Components:** Apart from all of these, there are some other components too that carry out a huge task in order to make Hadoop capable of processing large datasets. They are as follows:

* **Solr, Lucene:**These are the two services that perform the task of searching and indexing with the help of some java libraries, especially Lucene is based on Java which allows spell check mechanism, as well. However, Lucene is driven by Solr.
* **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, inter-component based communication, grouping, and maintenance.
* **Oozie:**Oozie simply performs the task of a scheduler, thus scheduling jobs and binding them together as a single unit. There is two kinds of jobs .i.e Oozie workflow and Oozie coordinator jobs. Oozie workflow is the jobs that need to be executed in a sequentially ordered manner whereas Oozie Coordinator jobs are those that are triggered when some data or external stimulus is given to it.