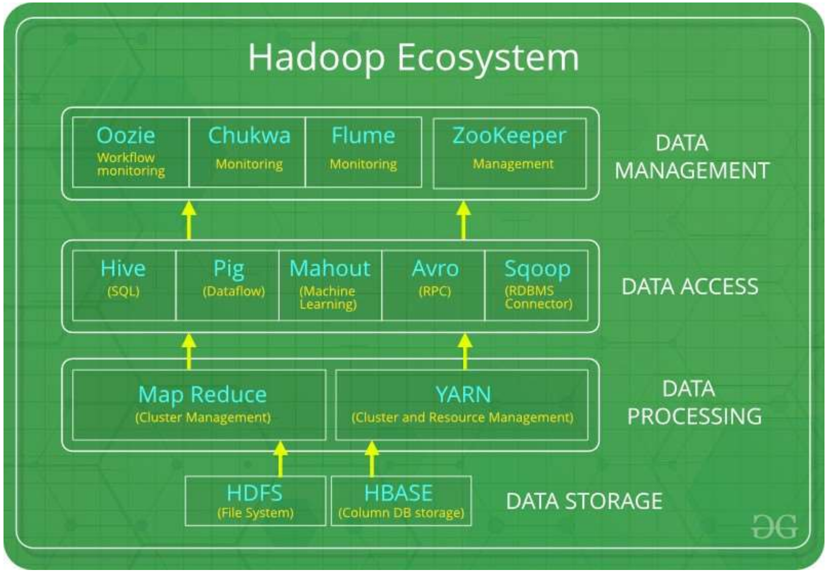
Experiment No.1
Hadoop HDFS Practical
Date of Performance:
Date of Submission:

AIM: Installation, Configuration of hadoop and performing basic file management operations in hadoop.

THEORY:

What is the Hadoop Ecosystem?

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.



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 algorithm libraries
- Solar, Lucene: Searching and Indexing
- Zookeeper: Managing cluster
- Oozie: Job Scheduling 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.

- Name node
- 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. HDFS maintains all the coordination between the clusters and hardware. 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.

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:

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

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. 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.

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.

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.

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.

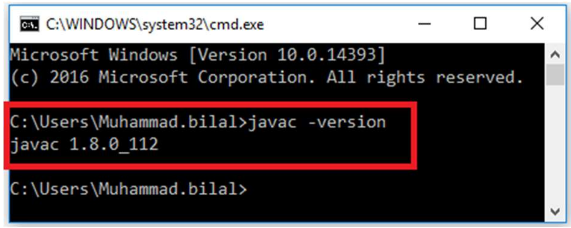
Steps in installing and configuring Hadoop:

Prepare:

1. Download Hadoop
2. Download Java JDK

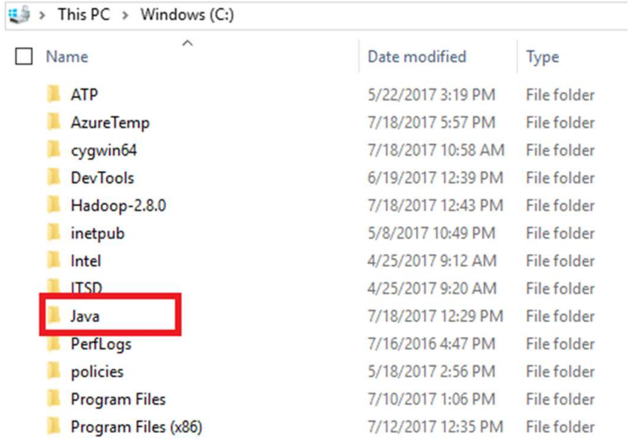
Setup:

1. Check either Java 1.8.0 is already installed on your system or not, use "Javac -version" to check.



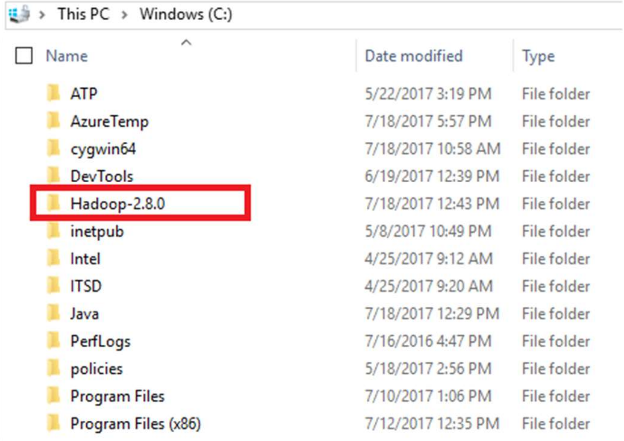
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Muhammad.bilal>javac -version
javac 1.8.0_112

C:\Users\Muhammad.bilal>
```

2. If Java is not installed on your system then first install java under "C:\JAVA"



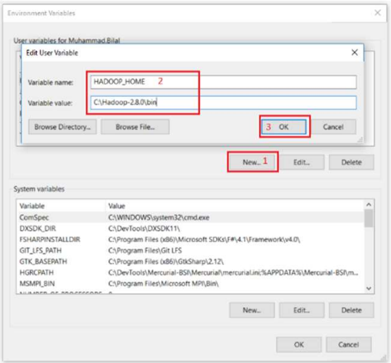
Name	Date modified	Type
ATP	5/22/2017 3:19 PM	File folder
AzureTemp	7/18/2017 5:57 PM	File folder
cygwin64	7/18/2017 10:58 AM	File folder
DevTools	6/19/2017 12:39 PM	File folder
Hadoop-2.8.0	7/18/2017 12:43 PM	File folder
inetpub	5/8/2017 10:49 PM	File folder
Intel	4/25/2017 9:12 AM	File folder
ITSD	4/25/2017 9:20 AM	File folder
Java	7/18/2017 12:29 PM	File folder
PerfLogs	7/16/2016 4:47 PM	File folder
policies	5/18/2017 2:56 PM	File folder
Program Files	7/10/2017 1:06 PM	File folder
Program Files (x86)	7/12/2017 12:35 PM	File folder

3. Extract file Hadoop 2.8.0.tar.gz or Hadoop-2.8.0.zip and place under "C:\Hadoop-2.8.0".



Name	Date modified	Type
ATP	5/22/2017 3:19 PM	File folder
AzureTemp	7/18/2017 5:57 PM	File folder
cygwin64	7/18/2017 10:58 AM	File folder
DevTools	6/19/2017 12:39 PM	File folder
Hadoop-2.8.0	7/18/2017 12:43 PM	File folder
inetpub	5/8/2017 10:49 PM	File folder
Intel	4/25/2017 9:12 AM	File folder
ITSD	4/25/2017 9:20 AM	File folder
Java	7/18/2017 12:29 PM	File folder
PerfLogs	7/16/2016 4:47 PM	File folder
policies	5/18/2017 2:56 PM	File folder
Program Files	7/10/2017 1:06 PM	File folder
Program Files (x86)	7/12/2017 12:35 PM	File folder

4. Set the path HADOOP_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).



Environment Variables

User variables for Muhammad.Bilal

Edit User Variable

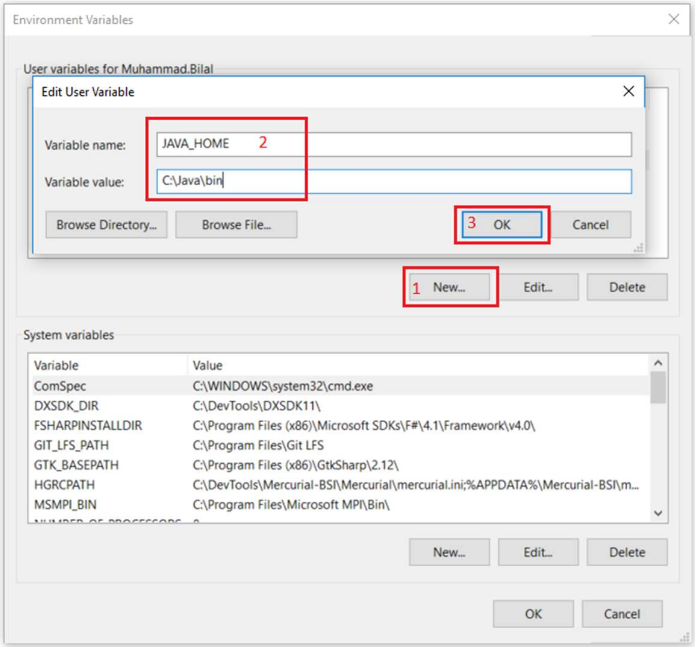
Variable name: HADOOP_HOME 2

Variable value: C:\Hadoop-2.8.0\bin

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DXSDK_DIR	C:\DevTools\DXSDK11\
FSHARPINSTALLDIR	C:\Program Files (x86)\Microsoft SDKs\F#\4.1\Framework\v4.0\
GIT_LFS_PATH	C:\Program Files\Git LFS
GTK_BASEPATH	C:\Program Files (x86)\GtkSharp\2.12\
HGRCPATH	C:\DevTools\Mercurial-BSI\Mercurial\mercurial.ini;%APPDATA%\Mercurial-BSI\m...
MSMPI_BIN	C:\Program Files\Microsoft MPI\Bin\

5. Set the path JAVA_HOME Environment variable on windows 10(see Step 1,2,3 and 4 below).



Environment Variables

User variables for Muhammad.Bilal

Edit User Variable

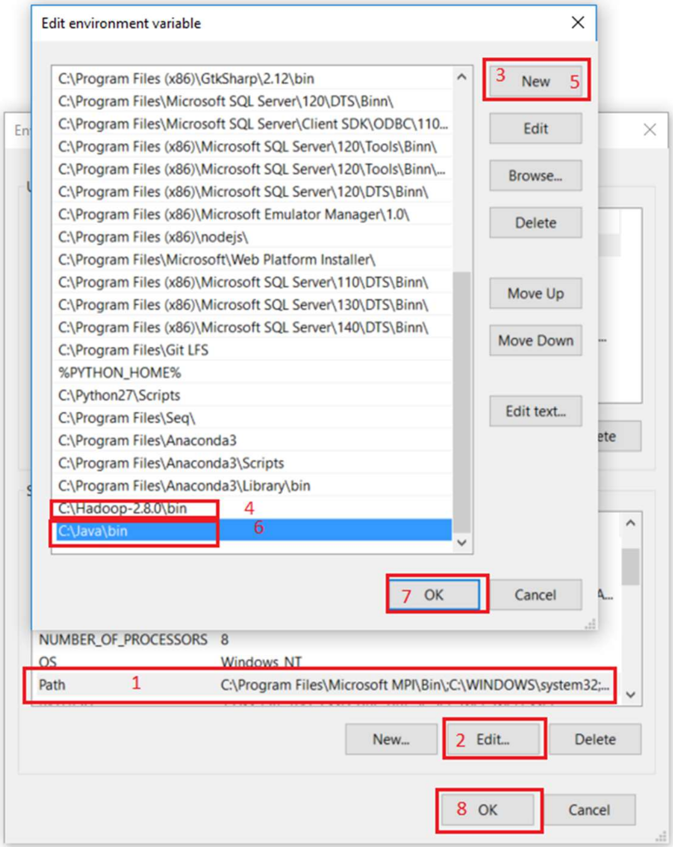
Variable name: JAVA_HOME 2

Variable value: C:\Java\bin

System variables

Variable	Value
ComSpec	C:\WINDOWS\system32\cmd.exe
DXSDK_DIR	C:\DevTools\DXSDK11\
FSHARPINSTALLDIR	C:\Program Files (x86)\Microsoft SDKs\F#\4.1\Framework\v4.0\
GIT_LFS_PATH	C:\Program Files\Git LFS
GTK_BASEPATH	C:\Program Files (x86)\GtkSharp\2.12\
HGRCPATH	C:\DevTools\Mercurial-BSI\Mercurial\mercurial.ini;%APPDATA%\Mercurial-BSI\m...
MSMPI_BIN	C:\Program Files\Microsoft MPI\Bin\

6. Next we set the Hadoop bin directory path and JAVA bin directory path.



Edit environment variable

C:\Program Files (x86)\GtkSharp\2.12\bin
C:\Program Files\Microsoft SQL Server\120\DTS\Binn\
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\110...
C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\
C:\Program Files (x86)\Microsoft SQL Server\120\Tools\Binn\...
C:\Program Files (x86)\Microsoft SQL Server\120\DTS\Binn\
C:\Program Files (x86)\Microsoft Emulator Manager\1.0\
C:\Program Files (x86)\nodejs\
C:\Program Files\Microsoft\Web Platform Installer\
C:\Program Files (x86)\Microsoft SQL Server\110\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\130\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\
C:\Program Files\Git LFS
%PYTHON_HOME%
C:\Python27\Scripts
C:\Program Files\Seq\
C:\Program Files\Anaconda3
C:\Program Files\Anaconda3\Scripts
C:\Program Files\Anaconda3\Library\bin
C:\Hadoop-2.8.0\bin 4
C:\Java\bin 6

NUMBER_OF_PROCESSORS 8
OS Windows_NT
Path 1 C:\Program Files\Microsoft MPI\Bin\;C:\WINDOWS\system32;...

Configuration:

1. Edit file C:/Hadoop 2.8.0/etc/hadoop/core-site.xml, paste below xml paragraph and save this file.

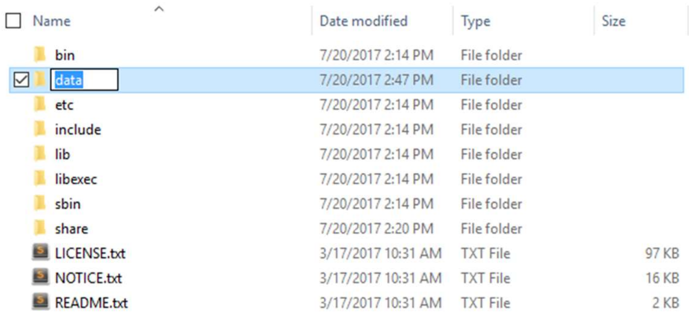
```
<configuration>
  <property>
     <name>fs.defaultFS</name>
     <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

2. Rename "mapred-site.xml.template" to "mapred-site.xml" and edit this file C:/Hadoop-2.8.0/etc/hadoop/mapred-site.xml, paste below xml paragraph and save this file.

```
<configuration>
  <property>
     <name>mapreduce.framework.name</name>
     <value>yarn</value>
  </property>
</configuration>
```

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

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



□ Name	Date modified	Type	Size
bin	7/20/2017 2:14 PM	File folder	
☑ data	7/20/2017 2:47 PM	File folder	
etc	7/20/2017 2:14 PM	File folder	
include	7/20/2017 2:14 PM	File folder	
lib	7/20/2017 2:14 PM	File folder	
libexec	7/20/2017 2:14 PM	File folder	
sbin	7/20/2017 2:14 PM	File folder	
share	7/20/2017 2:20 PM	File folder	
LICENSE.txt	3/17/2017 10:31 AM	TXT File	97 KB
NOTICE.txt	3/17/2017 10:31 AM	TXT File	16 KB
README.txt	3/17/2017 10:31 AM	TXT File	2 KB

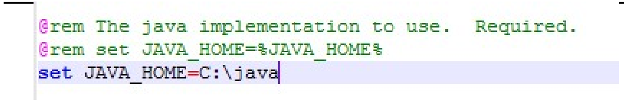
4. Edit file C:\Hadoop-2.8.0/etc/hadoop/hdfs-site.xml, paste below xml paragraph and save this file.

```
<configuration>
  <property>
     <name>dfs.replication</name>
     <value>1</value>
  </property>
  <property>
     <name>dfs.namenode.name.dir</name>
     <value>/hadoop-2.8.0/data/namenode</value>
  </property>
  <property>
     <name>dfs.datanode.data.dir</name>
     <value>/hadoop-2.8.0/data/datanode</value>
  </property>
</configuration>
```

5. Edit file C:/Hadoop-2.8.0/etc/hadoop/yarn-site.xml, paste below xml paragraph and save this file.

```
<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>
```

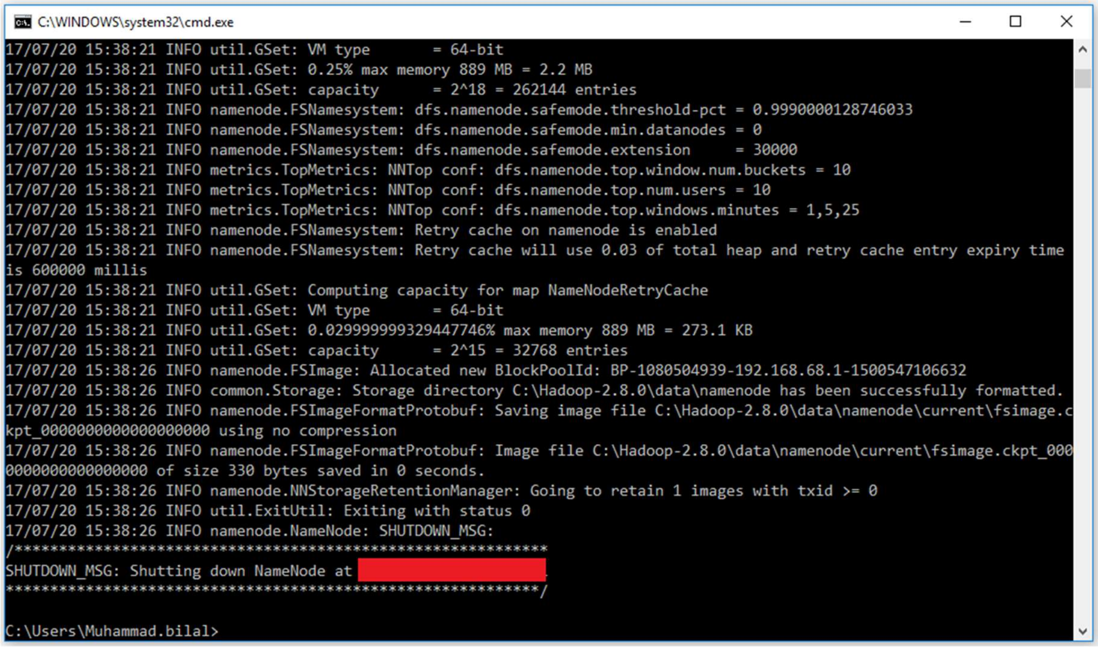
6. Edit file C:/Hadoop-2.8.0/etc/hadoop/hadoop-env.cmd by closing the command line "JAVA_HOME=%JAVA_HOME%" instead of set "JAVA_HOME=C:\Java" (On C:\java this is path to file jdk.18.0)



```
@rem The java implementation to use.  Required.
@rem set JAVA_HOME=%JAVA_HOME%
set JAVA_HOME=C:\java
```

Hadoop Configuration

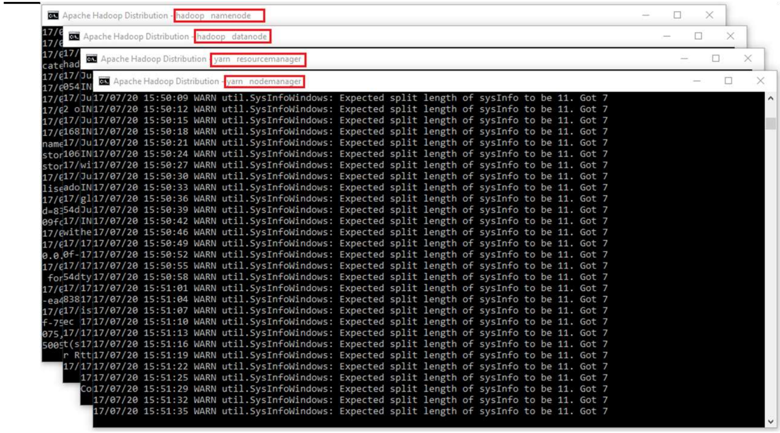
1. Download file Hadoop Configuration.zip
2. Delete file bin on C:\Hadoop-2.8.0\bin, replaced by file bin on file just download (from Hadoop Configuration.zip).
3. Open cmd and typing command "hdfs namenode –format".You will see



```
17/07/20 15:38:21 INFO util.GSet: VM type       = 64-bit
17/07/20 15:38:21 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
17/07/20 15:38:21 INFO util.GSet: capacity      = 2^18 = 262144 entries
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
17/07/20 15:38:21 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension     = 30000
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
17/07/20 15:38:21 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
17/07/20 15:38:21 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
17/07/20 15:38:21 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time
is 600000 millis
17/07/20 15:38:21 INFO util.GSet: Computing capacity for map NameNodeRetryCache
17/07/20 15:38:21 INFO util.GSet: VM type       = 64-bit
17/07/20 15:38:21 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
17/07/20 15:38:21 INFO util.GSet: capacity      = 2^15 = 32768 entries
17/07/20 15:38:26 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1080504939-192.168.68.1-1500547106632
17/07/20 15:38:26 INFO common.Storage: Storage directory C:\Hadoop-2.8.0\data\namenode has been successfully formatted.
17/07/20 15:38:26 INFO namenode.FSImageFormatProtobuf: Saving image file C:\Hadoop-2.8.0\data\namenode\current\fsimage.c
kpt_0000000000000000000 using no compression
17/07/20 15:38:26 INFO namenode.FSImageFormatProtobuf: Image file C:\Hadoop-2.8.0\data\namenode\current\fsimage.ckpt_000
0000000000000000 of size 330 bytes saved in 0 seconds.
17/07/20 15:38:26 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
17/07/20 15:38:26 INFO util.ExitUtil: Exiting with status 0
17/07/20 15:38:26 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at [redacted]
************************************************************/

C:\Users\Muhammad.bilal>
```

Testing

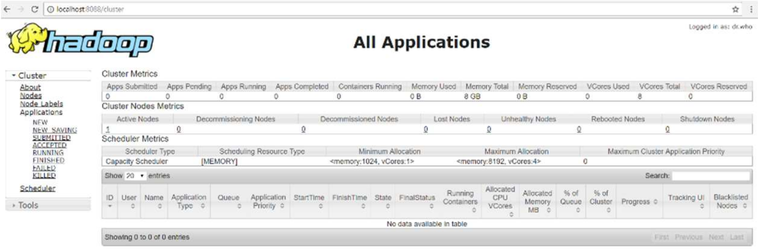
1. Open cmd and change directory to "C:\Hadoop-2.8.0\sbin" and type "start-all.cmd" to start apache.
2. Make sure these apps are running
   - Hadoop Namenode
   - Hadoop datanode
   - YARN Resourc Manager
   - YARN Node Manager



Apache Hadoop Distribution - hadoop namenode
Apache Hadoop Distribution - hadoop datanode
Apache Hadoop Distribution - yarn resourcemanager
Apache Hadoop Distribution - yarn nodemanager

```
17/0
17/0
17/017/
catehad
17/017/Ju
17/0054IN
17/017/Ju17/07/20 15:50:09 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/02 oIN17/07/20 15:50:12 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/Ju17/07/20 15:50:15 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/0168IN17/07/20 15:50:18 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
name17/Ju17/07/20 15:50:21 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
stor106IN17/07/20 15:50:24 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
stor17/wi17/07/20 15:50:27 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/Ju17/07/20 15:50:30 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
liseadoIN17/07/20 15:50:33 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/gl17/07/20 15:50:36 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
d=8354dJu17/07/20 15:50:39 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
09fc17/IN17/07/20 15:50:42 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/0withe17/07/20 15:50:46 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/1717/07/20 15:50:49 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
0.0.0f-1717/07/20 15:50:52 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/1717/07/20 15:50:55 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
 for54dty17/07/20 15:50:58 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/1717/07/20 15:51:01 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
-ea48381717/07/20 15:51:04 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
17/017/is17/07/20 15:51:07 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
f-75ec 1717/07/20 15:51:10 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
075,17/1717/07/20 15:51:13 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
5005t(s1717/07/20 15:51:16 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
    r Rtt17/07/20 15:51:19 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
    17/1717/07/20 15:51:22 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
       1717/07/20 15:51:25 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
       Co17/07/20 15:51:29 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
         17/07/20 15:51:32 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
         17/07/20 15:51:35 WARN util.SysInfoWindows: Expected split length of sysInfo to be 11. Got 7
```

3. Open: http://localhost:8088



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	8 GB	0 B	0	8	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
1	0	0	0	0	0	0

Scheduler Metrics

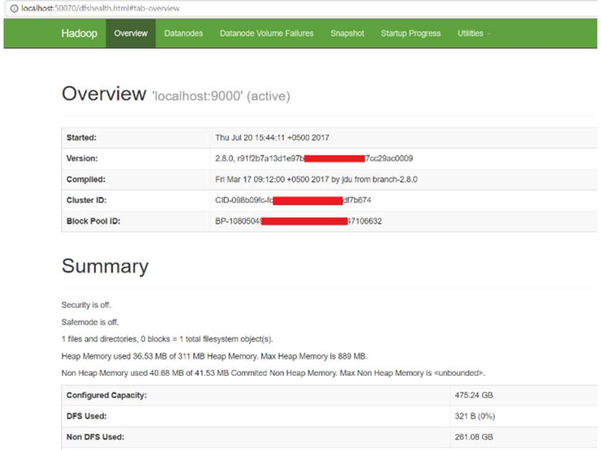
Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																	

Showing 0 to 0 of 0 entries First Previous Next Last

4. Open: http://localhost:50070



Overview 'localhost:9000' (active)

Started:	Thu Jul 20 15:44:11 +0500 2017
Version:	2.8.0, r91f2b7a13d1e97b[redacted]7cc29ac0009
Compiled:	Fri Mar 17 09:12:00 +0500 2017 by jdu from branch-2.8.0
Cluster ID:	CID-098b09fc-fc[redacted]df7b674
Block Pool ID:	BP-1080504[redacted]47106632

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 36.53 MB of 311 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 40.68 MB of 41.53 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	475.24 GB
DFS Used:	321 B (0%)
Non DFS Used:	261.08 GB

File management tasks in hadoop
In order to perform operations on Hadoop like copy, delete, move etc., following steps can be used:
Basic operations:

1. Create a directory in HDFS at given path(s). Usage:
hadoop fs -mkdir <paths>

2. List the contents of a directory. Usage :
hadoop fs -ls <args>

3. See contents of a file Same as unix cat command:
Usage:
hadoop fs -cat <path[filename]>
4. Copy a file from source to destination
This command allows multiple sources as well in which case the destination must be a directory.
Usage:
hadoop fs -cp <source> <dest>
5. Copy a file from/To Local file system to HDFS copyFromLocal
Usage:
hadoop fs -copyFromLocal <localsrc> URI
Similar to put command, except that the source is restricted to a local file reference.
copyToLocal
Usage:
hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>
Similar to get command, except that the destination is restricted to a local file reference.
7. Move file from source to destination.
Note:- Moving files across filesystem is not permitted. Usage :
hadoop fs -mv <src> <dest>
8. Remove a file or directory in HDFS.
Remove files specified as argument. Deletes directory only when it is empty Usage :
hadoop fs -rm <arg>

Steps for copying file
1) Go to Hadoop folder and then to sbin C:\>cd C:\hadoop-2.8.0\sbin
2) Start namenode and datanode with this command, Two more cmd windows will open C:\hadoop-2.8.0\sbin>start-dfs.cmd
3) Now start yarn through following command, Two more windows will open, one for yarn resource manager and one for yarn node manager
C:\hadoop-2.8.0\sbin>start-yarn.cmd
4) Create a directory named 'sample' in the hadoop directory using the following command C:\hadoop-2.8.0\sbin> hdfs dfs -mkdir /sample
5) To verify if the directory is created C:\hadoop-2.8.0\sbin>hdfs dfs -ls /
6) Copy text file from D drive to sample
C:\hadoop-2.8.0\sbin>hdfs dfs -copyFromLocal d:\rally.txt /sample
7) To verify if the file is copied C:\hadoop-2.8.0\sbin>hdfs dfs -ls /sample

CONCLUSION:

Explain two Functions of MapReduce.

1. Map Function
   - Purpose: The Map function processes input data and generates key-value pairs as output.
   - How it works:
     - The Map function takes a set of input data, typically stored in a distributed file system.
     - It processes each data item (e.g., a line in a text file) and outputs a set of key-value pairs.
     - For example, in a word count operation, the Map function would output each word in the input as a key with a value of 1 ((word, 1)).
     - These key-value pairs are then passed to the next stage, which is the shuffle and sort phase, where they are grouped by key.
2. Reduce Function
   - Purpose: The Reduce function aggregates or summarizes the output of the Map function.
   - How it works:
     - The Reduce function takes the grouped key-value pairs from the shuffle and sort phase.
     - For each unique key, it processes all associated values, typically performing an aggregation operation (e.g., summing values).
     - For example, in the word count operation, the Reduce function would sum all the values for each word key, producing an output like (word, total_count).
     - The final output is a reduced set of data that is often stored back in the distributed file system.