

DATA ANALYTICS

ASSIGNMENT 1

MADHUMITHAS

210701142

CSE C

INTRODUCTION ABOUT HADOOP:

1.1 HISTORY OF HADOOP:

Hadoop originated from the need to process large-scale data across distributed systems. In the early 2000s, Google published papers on the Google File System (GFS) and MapReduce, which inspired Doug Cutting and Mike Cafarella. They were working on the Nutch project, a search engine, and soon realized the need for a robust framework to handle massive datasets. In 2005, Hadoop was born, named after Cutting's son's toy elephant, with its core components: HDFS (Hadoop Distributed File System) and the MapReduce programming model.

Yahoo! was one of the earliest adopters of Hadoop and contributed significantly to its development. By 2008, Hadoop had become a top-level Apache project, and Yahoo! set a record by using it to process 1 terabyte of data in 209 seconds, showcasing its potential for large-scale data processing.

Hadoop's popularity led to the development of an ecosystem around it, including projects like Apache Pig, Hive, and HBase, which further extended its capabilities for data processing, storage, and analysis. Hadoop became the foundation of big data processing and has been widely adopted across various industries for handling vast amounts of data.

1.2 VERSIONS OF HADOOP:

1. Hadoop 0.x Series

Hadoop 0.1.0 (2006): The initial release when Hadoop was still a subproject of Apache Lucene.

Hadoop 0.18.0 (2008): Introduced the first version of Hadoop as a top-level Apache project. This version included a stable MapReduce framework and HDFS (Hadoop Distributed File System).

2. Hadoop 1.x Series (2009 - 2012)

Hadoop 1.0.0 (2011): Marked the first major release of Hadoop, featuring a stable MapReduce framework and HDFS. This version was widely adopted for production workloads.

Hadoop 1.2.x: Improved stability and performance of the Hadoop 1.x series. Hadoop 1.x was built around the original MapReduce and HDFS components.

3. Hadoop 2.x Series (2012 - 2017)

Hadoop 2.0.0 (2012): Introduced YARN (Yet Another Resource Negotiator), which decoupled the resource management and job scheduling functions from MapReduce, allowing Hadoop to support other distributed processing models besides MapReduce.

Hadoop 2.2.0 (2013): Officially marked Hadoop 2 as stable, with YARN becoming the default resource management layer. This version also introduced HDFS Federation, improving scalability.

Hadoop 2.7.x (2015): Introduced significant improvements, including enhanced HDFS encryption, resource-aware scheduling in YARN, and improved usability for Hadoop as a whole.

Hadoop 2.8.x (2017): Focused on performance improvements, bug fixes, and enhancements, with ongoing support and maintenance for users of the 2.x line.

4. Hadoop 3.x Series (2017 - Present)

Hadoop 3.0.0 (2017): A major release introducing several new features, including support for erasure coding in HDFS, which reduces storage overhead, and multiple NameNodes for high availability. It also brought Docker container support in YARN, allowing for better resource utilization and isolation.

Hadoop 3.1.x (2018): Added native support for GPUs in YARN, allowing for better support of machine learning workloads. This version also introduced intra-node disk balancing.

Hadoop 3.2.x (2019): Introduced further stability improvements, including improvements to HDFS, YARN, and the integration of newer technologies like TensorFlow.

Hadoop 3.3.x (2020): Continued to build on the stability and performance improvements, along with better cloud integration, including S3A and Azure Blob storage enhancements.

5. Hadoop 4.x Series (Future)

As of the last update in 2023, the Hadoop 4.x series has not yet been released, but it is expected to continue the evolution of Hadoop, focusing on cloud-native features, improved machine learning support, and even greater scalability and performance enhancements.

1.3 SYSTEM REQUIREMENTS:

1. Hardware Requirements

Memory (RAM):

Minimum: 4 GB per node for small, non-production clusters.

Recommended: 8 GB to 64 GB per node, depending on the workload. High-memory configurations are preferred for heavy data processing tasks.

CPU:

Minimum: Dual-core processors per node.

Recommended: Quad-core processors or higher with multiple cores per node. The number of cores should scale with the expected workload.

Storage:

Minimum: 1 TB of disk space per node.

Recommended: Several terabytes per node with additional disks for data storage. Hadoop works best with a large amount of disk space to store data across the HDFS.

Disk Type: Use SATA/SAS HDDs for storage; SSDs can be used for better performance, particularly for intermediate data processing.

Network:

Minimum: 1 Gbps network connectivity between nodes.

Recommended: 10 Gbps network connectivity for better performance in large clusters.

2. Operating System

Supported OS:

Hadoop primarily runs on Linux distributions (e.g., CentOS, Ubuntu, Red Hat Enterprise Linux).

It can run on Windows, but Linux is strongly recommended for production environments.

Ensure the system is 64-bit as Hadoop requires a 64-bit OS.

File System:

The underlying file system should support extended attributes, and the local file system should be configured for high I/O performance.

3. Java Requirements

Java Version: Minimum: Java 8 (JDK 1.8)

Recommended: Java 11 or later, depending on the Hadoop version. Check compatibility with the Hadoop release you are using.

Java Home: Ensure that the JAVA_HOME environment variable is correctly set on all nodes.

4. Hadoop Version Requirements

Ensure that the Hadoop version you choose is compatible with your OS, Java version, and any other software components you plan to use (e.g., Hive, Spark, etc.).

Different versions may have additional software dependencies, so review the release notes for the version you plan to install.

5. Additional Software Requirements

SSH: Password-less SSH access is required between all nodes in the cluster for Hadoop to operate.

Python: Some Hadoop tools or auxiliary programs may require Python, usually Python 2.x or Python 3.x, depending on the Hadoop version.

6. Cluster Setup Considerations

Node Types: Typically, you will have master nodes (NameNode, ResourceManager) and worker nodes (DataNode, NodeManager). The hardware requirements for master nodes may differ, often requiring more RAM and CPU resources than worker nodes.

High Availability (HA): For production environments, consider setting up NameNode and ResourceManager in high availability mode, which may require additional nodes.

7. Virtualization/Cloud Considerations

If deploying on a virtualized environment or in the cloud (AWS, Azure, Google Cloud), ensure that the virtual machines or cloud instances meet the minimum requirements and that network latency is minimized. Cloud-native Hadoop deployments should consider using cloud storage (e.g., Amazon S3, Google Cloud Storage) integrated with HDFS.

8. Monitoring and Management Tools

Tools: Install monitoring tools (e.g., Ganglia, Nagios) and management tools (e.g., Apache Ambari, Cloudera Manager) to monitor the health of the cluster and manage configurations easily.

1.4 INSTALLATION AND COMMANDS

Step 1: Download and install Java

Hadoop is built on Java, so you must have Java installed on your PC. You can get the most recent version of Java from the official website. After downloading, follow the installation wizard to install Java on your system.

JDK: <https://www.oracle.com/java/technologies/javase-downloads.html>

Step 2: Download Hadoop

Hadoop can be downloaded from the Apache Hadoop website. Make sure to have the latest stable release of Hadoop. Once downloaded, extract the contents to a convenient location.

Hadoop: <https://hadoop.apache.org/releases.html>

Step 3: Set Environment Variables

You must configure environment variables after downloading and unpacking Hadoop. Launch the Start menu, type “Edit the system environment variables,” and select the result. This will launch the System Properties dialogue box. Click on “Environment Variables” button to open.

Click “New” under System Variables to add a new variable. Enter the variable name “HADOOP_HOME” and the path to the Hadoop folder as the variable value. Then press “OK.”

Then, under System Variables, locate the “Path” variable and click “Edit.” Click “New” in the Edit Environment Variable window and enter “%HADOOP_HOME%bin” as the variable value. To close all the windows, use the “OK” button.

Step 4: Setup Hadoop

You must configure Hadoop in this phase by modifying several configuration files. Navigate to the “etc/hadoop” folder in the Hadoop folder. You must make changes to three files:

core-site.xml

hdfs-site.xml

mapred-site.xml

Open each file in a text editor and edit the following properties:

In core-site.xml

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

In hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/hadoop-3.3.1/data/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/hadoop-3.3.1/data/datanode</value>
  </property>
</configuration>
```

In mapred-site.xml

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:54311</value>
  </property>
</configuration>
```

Save the changes in each file.

Step 5: Format Hadoop NameNode

You must format the NameNode before you can start Hadoop. Navigate to the Hadoop bin folder using a command prompt. Execute this command:

```
hadoop namenode -format
```

Step 6: Start Hadoop

To start Hadoop, open a command prompt and navigate to the Hadoop bin folder. Run the following command:

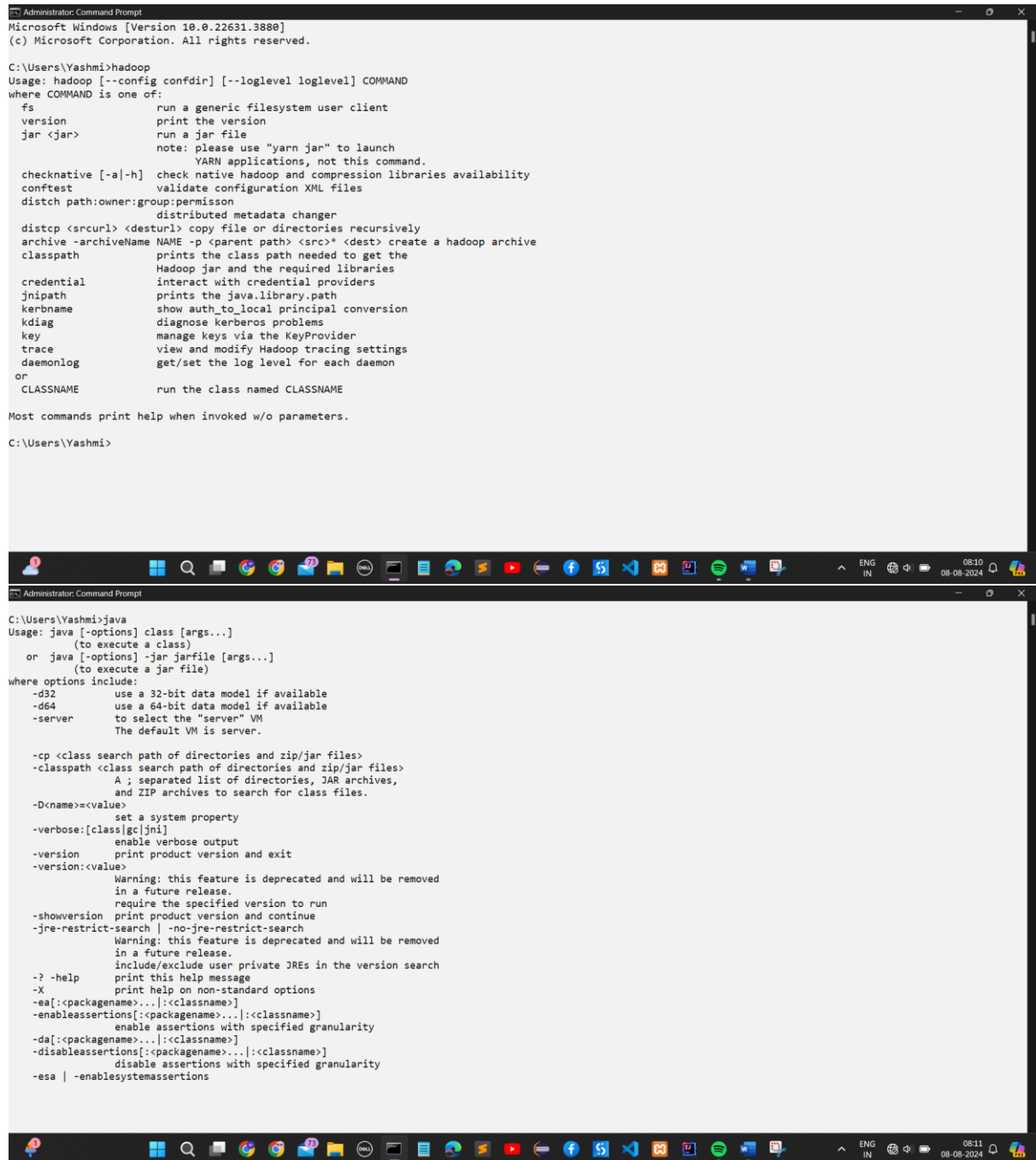
```
start-all.cmd
```

This command will start all the required Hadoop services, including the NameNode, DataNode, and JobTracker. Wait for a few minutes until all the services are started.

Step 7: Verify Hadoop Installation

To ensure that Hadoop is properly installed, open a web browser and go to <http://localhost:50070/>. This will launch the web interface for the Hadoop NameNode. You should see a page with Hadoop cluster information.

1.5 SCREENSHOTS:



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

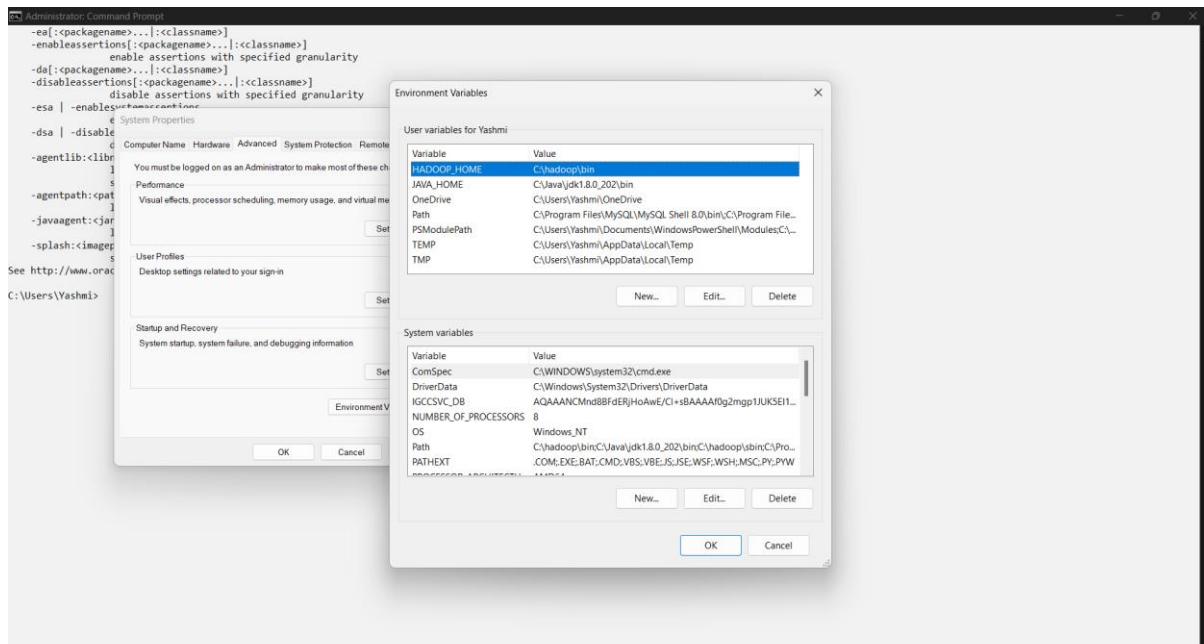
C:\Users\Yashmi>hadoop
Usage: hadoop [--config confdir] [--loglevel loglevel] COMMAND
where COMMAND is one of:
  fs                run a generic filesystem user client
  version           print the version
  jar <jar>         run a jar file
                   note: please use "yarn jar" to launch
                   YARN applications, not this command.
  checknative [-a|-h] check native hadoop and compression libraries availability
  conftest          validate configuration XML files
  distch path:owner:group:permission distributed metadata changer
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src> <dest> create a hadoop archive
  classpath          prints the class path needed to get the
                   Hadoop jar and the required libraries
  credential         interact with credential providers
  jnipath            prints the java.library.path
  kerbname           show auth_to_local principal conversion
  kdiag             diagnose kerberos problems
  key               manage keys via the KeyProvider
  trace             view and modify Hadoop tracing settings
  daemonlog          get/set the log level for each daemon
  or
  CLASSNAME          run the class named CLASSNAME

Most commands print help when invoked w/o parameters.

C:\Users\Yashmi>
```

```
Administrator: Command Prompt
C:\Users\Yashmi>java
Usage: java [-options] class [args...]
           (to execute a class)
  or  java [-options] -jar jarfile [args...]
       (to execute a jar file)
where options include:
  -d32          use a 32-bit data model if available
  -d64          use a 64-bit data model if available
  -server       to select the "server" VM
                 The default VM is server.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
               A ; separated list of directories, JAR archives,
               and ZIP archives to search for class files.
  -Dname=value  set a system property
  -verbose:[class|gc|jni]
               enable verbose output
  -version      print product version and exit
  -version:<value>
               Warning: this feature is deprecated and will be removed
               in a future release.
               require the specified version to run
  -showversion  print product version and continue
  -jre-restrict-search | -no-jre-restrict-search
               Warning: this feature is deprecated and will be removed
               in a future release.
               include/exclude user private JREs in the version search
  -? -help     print this help message
  -X           print help on non-standard options
  -ea[:<packagename>...]:<classname>]
               enable assertions with specified granularity
  -da[:<packagename>...]:<classname>]
               disable assertions with specified granularity
  -esa | -enablesystemassertions
               enable system assertions
  -esa | -disablesystemassertions
               disable system assertions
```

```

Administrator: Command Prompt
-ef[:packagename...[:<classname>]
-enableassertions[:<packagename...[:<classname>]
enable assertions with specified granularity
-da[:<packagename...[:<classname>]
-disableassertions[:<packagename...[:<classname>]
disable assertions with specified granularity
-esa | -enableassertions[:<packagename...[:<classname>]
enable assertions with specified granularity
-dsa | -disableassertions[:<packagename...[:<classname>]
disable assertions with specified granularity
System Properties
Computer Name Hardware Advanced System Protection Remote
You must be logged on as an Administrator to make most of these ch
Performance
Visual effects, processor scheduling, memory usage, and virtual me
User Profiles
Desktop settings related to your sign
Startup and Recovery
System startup, system failure, and debugging information
Environment Variables
OK Cancel

Environment Variables
User variables for Yashmi
Variable Value
HADOOP_HOME C:\hadoop\bin
JAVA_HOME C:\Java\jdk1.8.0_202\bin
OneDrive C:\Users\Yashmi\OneDrive
Path C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Program File...
PSModulePath C:\Users\Yashmi\Documents\WindowsPowerShell\Modules\C\...
TEMP C:\Users\Yashmi\AppData\Local\Temp
TMP C:\Users\Yashmi\AppData\Local\Temp
New... Edit... Delete
OK Cancel

System variables
Variable Value
ComSpec C:\WINDOWS\system32\cmd.exe
DriverData C:\Windows\System32\Drivers\DriverData
IGCCSV_C_DB AQAAANCMnd8BFdRJHoAwE/Ci+sBAAAf0g2mgp1JUKSE1...
NUMBER_OF_PROCESSORS 8
OS Windows_NT
Path C:\hadoop\bin\;C:\Java\jdk1.8.0_202\bin\;C:\hadoop\bin\;C\Pro...
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.P...
New... Edit... Delete
OK Cancel

Administrator: Command Prompt
disable system assertions
-agentlib:<libname>[=<options>]
load native agent library <libname>, e.g. -agentlib:hprof
see also, -agentlib:jvdp=help and -agentlib:hprof=help
-agentpath:<pathname>[=<options>]
load native agent library by full pathname
-javaagent:<jarpath>[=<options>]
load Java programming language agent, see java.lang.instrument
-splash:<imagepath>
show splash screen with specified image
See http://www.oracle.com/technetwork/java/javase/documentation/index.html for more details.

C:\Users\Yashmi>jps
6016 Jps

C:\Users\Yashmi>start-dfs.cmd

C:\Users\Yashmi>cd /

C:\>cd hadoop

C:\hadoop>cd sbin

C:\hadoop\sbin>start-dfs.cmd

C:\hadoop\sbin>jps
2020 Jps
13852 DataNode
19788 NameNode

```

Navbar

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Overview 'localhost:9000' (✓active)

Started:	Thu Aug 08 08:16:28 +0530 2024
Version:	3.3.6, r1b678238728da9269a4f88195058f08d012b9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-254213b3-a469-4277-84ff-b2c1b07303
Block Pool ID:	BP-315754006-192.168.56.1-1722840223172

Summary

Security is off.
Safemode is off.
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).
Heap Memory used 109.78 MB of 199 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 50.06 MB of 51.31 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	219.01 GB
Configured Remote Capacity:	0 B
DFS Used:	320 B (0%)

Navbar

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Last Checkpoint Time

Thu Aug 08 08:16:30 +0530 2024

Enabled Erasure Coding Policies

RS-6-3-1024k

NameNode Journal Status

Current transaction ID: 2

Journal Manager	State
FileJournalManager(root=/tmp/hadoop-Yashmi/dfs/name)	EditLogOutputStream(/tmp/hadoop-Yashmi/dfs/name/current/edits_inprogress_0000000000000000002)

NameNode Storage

Storage Directory	Type	State
/tmp/hadoop-Yashmi/dfs/name	IMAGE_AND_EDITS	Active

DFS Storage Types

Storage Type	Configured Capacity	Capacity Used	Capacity Remaining	Block Pool Used	Nodes In Service
DISK	219.01 GB	320 B (0%)	5.31 GB (2.43%)	320 B	1

Hadoop, 2023.

