CS19P16 – Data Analytics Roll No: 210701162

# Assignment 1 – Apache Hadoop

## Introduction:

Apache Hadoop is an open-source framework designed to process large datasets across distributed clusters of computers using simple programming models. It scales from a single server to thousands of machines, each providing both local computation and storage. Hadoop is extensively utilized for big data analytics and supports a variety of applications, including data warehousing, ETL (Extract, Transform, Load) processes, machine learning, and data mining.

## Evolution of Hadoop:

Hadoop's development stemmed from projects at Yahoo! and Apache Nutch, drawing inspiration from Google's publications on the Google File System (GFS) and MapReduce.

- **2003:** Google released the GFS paper, outlining a distributed file system designed to manage large datasets using commodity hardware.
- **2004:** Google published the MapReduce paper, which introduced a programming model and implementation for processing and generating large datasets.
- **2006:** Doug Cutting and Mike Cafarella developed Hadoop to support the Nutch search engine project.
- **2008:** Hadoop was elevated to a top-level Apache project, leading to broader adoption and continued development.
- Since then, Hadoop has become a fundamental technology in big data, benefiting from ongoing contributions and enhancements from both the open-source community and industry.

## Hadoop Versions:

Hadoop has undergone several major updates, each bringing performance enhancements, new features, and broader support.

- **Hadoop 1.x:** The original version, which used MapReduce as its primary processing engine and HDFS for storage.
- **Hadoop 2.x:** Introduced YARN (Yet Another Resource Negotiator) for improved resource management and support for various processing models beyond MapReduce, such as Apache Tez and Apache Spark.

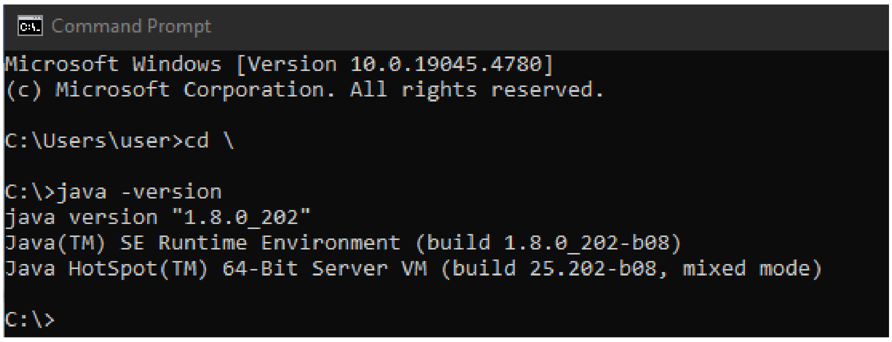
- **Hadoop 3.x:** Added features like erasure coding for more efficient storage, support for multiple Name Nodes, and integration with Docker for containerization.

**System Requirements:**
Hadoop can be installed on various operating systems, including Windows, macOS, and Linux. Here are the general system requirements:

- **RAM:** Minimum of 4 GB (8 GB or more recommended).
- **CPU:** Multi-core processor (quad-core or higher recommended for better performance).
- **Disk Space:** At least 20 GB of free space (more is preferable for handling large datasets and enhancing performance).
- **Java:** Requires Java Development Kit (JDK) version 8 or higher.
- **Operating Systems:**
  - **Linux:** Preferred for production clusters due to its stability and efficiency in distributed environments.
  - **macOS:** Suitable for development purposes.
  - **Windows:** Typically used in standalone mode rather than for production clusters.

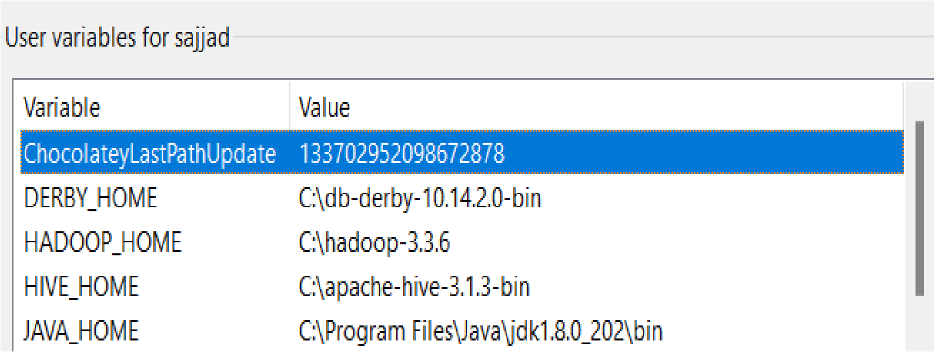
**Installation Steps:**

1. Install Java 1.8 version



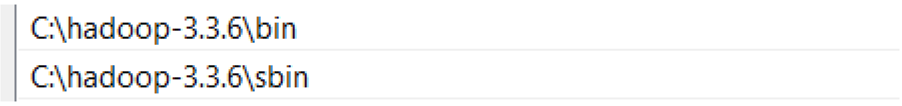
```
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd \

C:\>java -version
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)

C:\>
```

2. Download Hadoop 3.3.6 and set the environment variables.



User variables for sajjad

| Variable | Value |
|---|---|
| ChocolateyLastPathUpdate | 133702952098672878 |
| DERBY_HOME | C:\db-derby-10.14.2.0-bin |
| HADOOP_HOME | C:\hadoop-3.3.6 |
| HIVE_HOME | C:\apache-hive-3.1.3-bin |
| JAVA_HOME | C:\Program Files\Java\jdk1.8.0_202\bin |



```
C:\hadoop-3.3.6\bin
C:\hadoop-3.3.6\sbin
```

3. Configure the following files:

**core-site.xml:**

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

**hdfs-site.xml:**

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///C:/hadoop-3.3.6/data/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:///C:/hadoop-3.3.6/data/datanode</value>
</property>
</configuration>
```

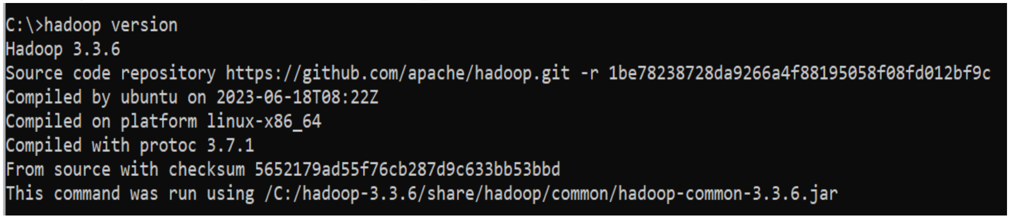
**mapred-site.xml:**

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

**yarn-site.xml:**

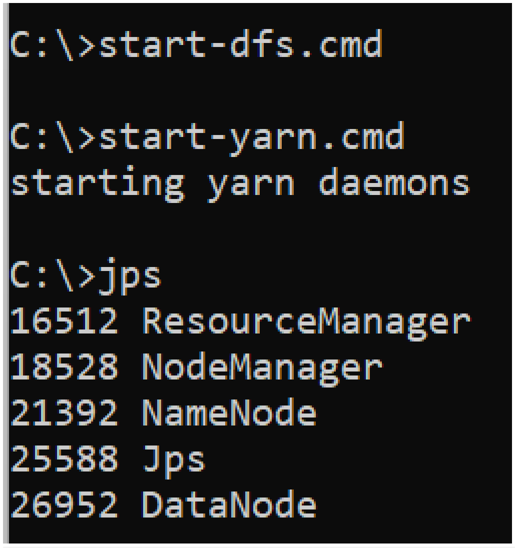
```
<?xml version="1.0"?>
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
</configuration>
```

4. Verify the version of Hadoop

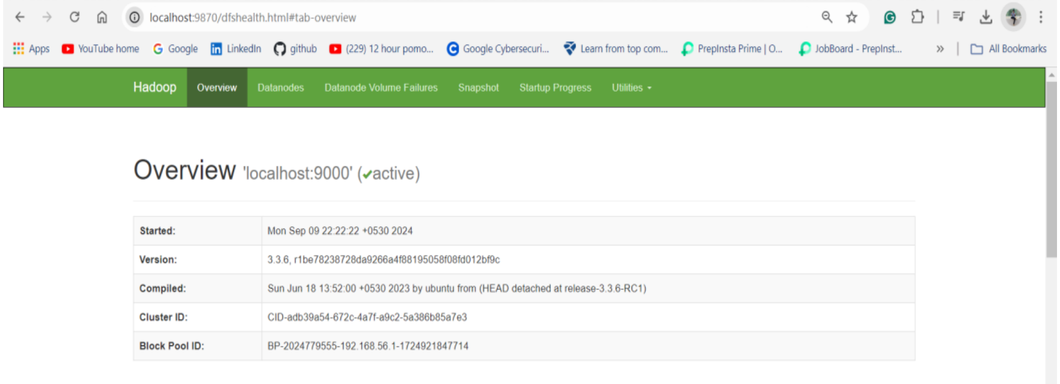


```
C:\>hadoop version
Hadoop 3.3.6
Source code repository https://github.com/apache/hadoop.git -r 1be78238728da9266a4f88195058f08fd012bf9c
Compiled by ubuntu on 2023-06-18T08:22Z
Compiled on platform linux-x86_64
Compiled with protoc 3.7.1
From source with checksum 5652179ad55f76cb287d9c633bb53bbd
This command was run using /C:/hadoop-3.3.6/share/hadoop/common/hadoop-common-3.3.6.jar
```

5. Start Hadoop using **start-dfs.cmd** and **start-yarn.cmd**. Verify that Hadoop has started using the **jps** command.



```
C:\>start-dfs.cmd

C:\>start-yarn.cmd
starting yarn daemons

C:\>jps
16512 ResourceManager
18528 NodeManager
21392 NameNode
25588 Jps
26952 DataNode
```



Overview 'localhost:9000' (✓active)

| | |
|---|---|
| **Started:** | Mon Sep 09 22:22:22 +0530 2024 |
| **Version:** | 3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c |
| **Compiled:** | Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1) |
| **Cluster ID:** | CID-adb39a54-672c-4a7f-a9c2-5a386b85a7e3 |
| **Block Pool ID:** | BP-2024779555-192.168.56.1-1724921847714 |