CS19P16 DATA ANALYTICS

Hadoop Installation

Introduction To Hadoop

Hadoop is an open source software programming framework for storing a large amount of data and performing the computation. Its framework is based on Java programming with some native code in C and shell scripts.

Hadoop is an open-source software framework that is used for storing and processing large amounts of data in a distributed computing environment. It is designed to handle big data and is based on the MapReduce programming model, which allows for the parallel processing of large datasets.

Hadoop is an open source software programming framework for storing a large amount of data and performing the computation. Its framework is based on Java programming with some native code in C and shell scripts.

Hadoop is an open-source software framework that is used for storing and processing large amounts of data in a distributed computing environment. It is designed to handle big data and is based on the MapReduce programming model, which allows for the parallel processing of large datasets.

History Of Hadoop:

The Hadoop was started by Doug Cutting and Mike Cafarella in 2002. Its origin was the Google File System paper, published by Google.

- In 2002, Doug Cutting and Mike Cafarella started to work on a project, **Apache Nutch.** It is an open source web crawler software project.
- While working on Apache Nutch, they were dealing with big data. To store that data they have to spend a lot of costs which becomes the consequence of that project. This problem becomes one of the important reason for the emergence of Hadoop.
- In 2003, Google introduced a file system known as GFS (Google file system). It is a proprietary distributed file system developed to provide efficient access to data.
- In 2004, Google released a white paper on Map Reduce. This technique simplifies the data processing on large clusters.
- In 2005, Doug Cutting and Mike Cafarella introduced a new file system known as NDFS (Nutch Distributed File System). This file system also includes Map reduce.

- In 2006, Doug Cutting quit Google and joined Yahoo. On the basis of the Nutch project, Dough Cutting introduces a new project Hadoop with a file system known as HDFS (Hadoop Distributed File System). Hadoop first version 0.1.0 released in this year.
- Doug Cutting gave named his project Hadoop after his son's toy elephant.
- In 2007, Yahoo runs two clusters of 1000 machines.
- In 2008, Hadoop became the fastest system to sort 1 terabyte of data on a 900 node cluster within 209 seconds.
- In 2013, Hadoop 2.2 was released.
- In 2017, Hadoop 3.0 was released.

Versions of Hadoop:

Apache Hadoop has gone through various versions over the years, with each version introducing new features, improvements, and bug fixes. Below is a summary of some key versions of Hadoop:

1. Hadoop 0.x Series

- **Initial Releases (0.1.0 to 0.20.x)**:
  - These versions were the early iterations of Hadoop, starting from the project's inception.
  - Hadoop 0.20.x was one of the more stable and widely used versions in the early days, laying the foundation for Hadoop's growth.

2. Hadoop 1.x Series

- **Hadoop 1.0.0 (December 2011)**:
  - Marked the first major stable release.
  - Included the core components: HDFS (Hadoop Distributed File System) and MapReduce.
  - The JobTracker and TaskTracker were used for job scheduling and resource management.
  - **Hadoop 1.2.x**: The last release in the 1.x series.

3. Hadoop 2.x Series

- **Hadoop 2.0.0 (October 2012)**:
  - Introduced YARN (Yet Another Resource Negotiator), a major architectural change that separated resource management from job scheduling, allowing for better resource utilization and support for multiple processing models.

- **Hadoop 2.2.0**: The first stable release in the 2.x series.
- **Hadoop 2.4.0 to 2.7.x**:
  - Introduced significant enhancements like HDFS Federation, high availability, and snapshots.
  - The 2.7.x series became very stable and widely adopted.
- **Hadoop 2.8.x and 2.9.x**: Continued improvements and backward compatibility.

4. Hadoop 3.x Series

- **Hadoop 3.0.0 (December 2017)**:
  - Introduced significant changes and new features like erasure coding for HDFS, a new resource management system, and support for more than two NameNodes.
  - Enhanced scalability, storage efficiency, and resource management.
  - **Hadoop 3.1.x to 3.3.x**:
    - Continued improvements, including HDFS Router-based Federation, improved scalability, and better cloud integration.
  - **Hadoop 3.3.0 (July 2020)**:
    - Focused on cloud storage integrations, including S3A enhancements for AWS S3, Azure Data Lake, and Google Cloud Storage.
  - **Hadoop 3.3.4 (January 2023)**:
    - The latest stable release with ongoing improvements, bug fixes, and enhancements, especially for cloud-native deployments.

System Requirements:

1. Operating Systems

- **Linux/Unix**:

Preferred and recommended OS for running Hadoop, as it was initially developed and tested on Linux.

Supported distributions include Ubuntu, CentOS, Red Hat, and others.

- **Windows**:

Hadoop can run on Windows, but it may require additional configurations. Windows 10/11 and Windows Server versions are commonly used.

- **macOS**:

Hadoop can be installed and run on macOS, mainly for development purposes. macOS Big Sur and later versions are supported.

2. Hardware Requirements

CPU:Minimum: 4-core processor.

Recommended: 8-core or higher, depending on workload.

Memory (RAM):

Minimum: 8 GB (for testing or small-scale deployments).

Recommended: 16 GB or higher (for production environments).

Memory requirements increase with the size of the Hadoop cluster and workload.

Storage:

Minimum: 10 GB of free disk space (for basic installation and testing).

Recommended: 100 GB or more, depending on data volume.

Distributed storage like HDFS requires significant disk space, often several terabytes.

Network:

Gigabit Ethernet recommended for cluster nodes.

Proper network configuration and low-latency are essential for a Hadoop cluster.

3. Software Requirements

- **Java**:

JDK Version: At least JDK 8 or later is required. Hadoop 3.3.x and later can run on JDK 8, 11, or higher.

JAVA_HOME: Set the JAVA_HOME environment variable to point to the installed JDK.

- **SSH**:

SSH must be configured on all nodes in the cluster, allowing password-less login for the Hadoop user.

Required for managing Hadoop daemons across the cluster.

- **Python**:

Python 2.7 or 3.x is required for some scripts and tools used by Hadoop.

- **Linux/Unix Dependencies**:

For Linux/Unix, ensure you have the following packages installed: ssh, rsync, curl, wget, tar, gzip, libc, libssl, and others depending on the specific Linux distribution.

- **Windows Dependencies**:

Windows-specific setups may require Cygwin or WSL (Windows Subsystem for Linux) for better compatibility.

Ensure you have a compatible version of WinRAR or 7-Zip for managing archives.

Installation steps:

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>C:/hadoop/hadoop/data/namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>C:/hadoop/hadoop/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:

hdfs 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-dfs.cmd

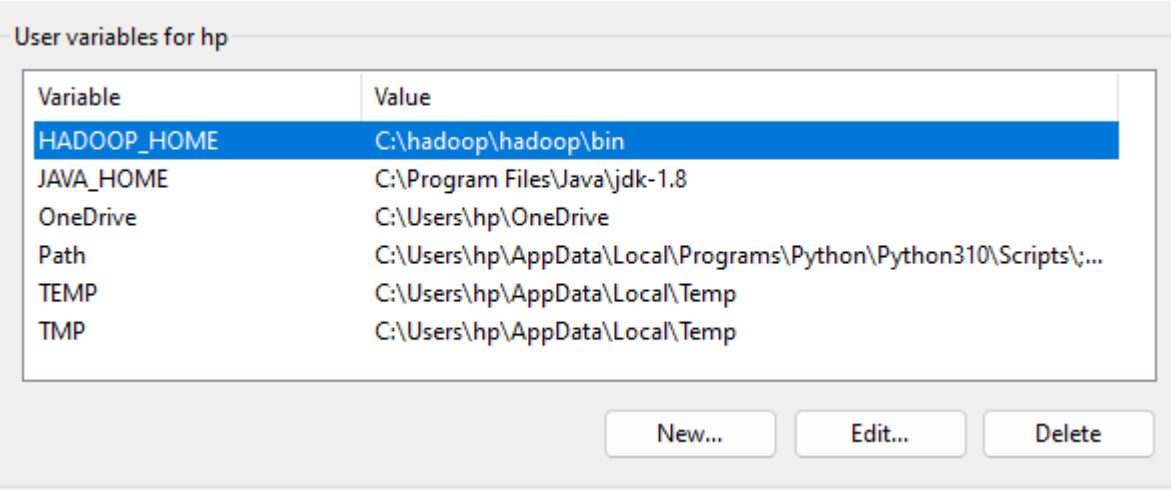
start-yarn.cmd

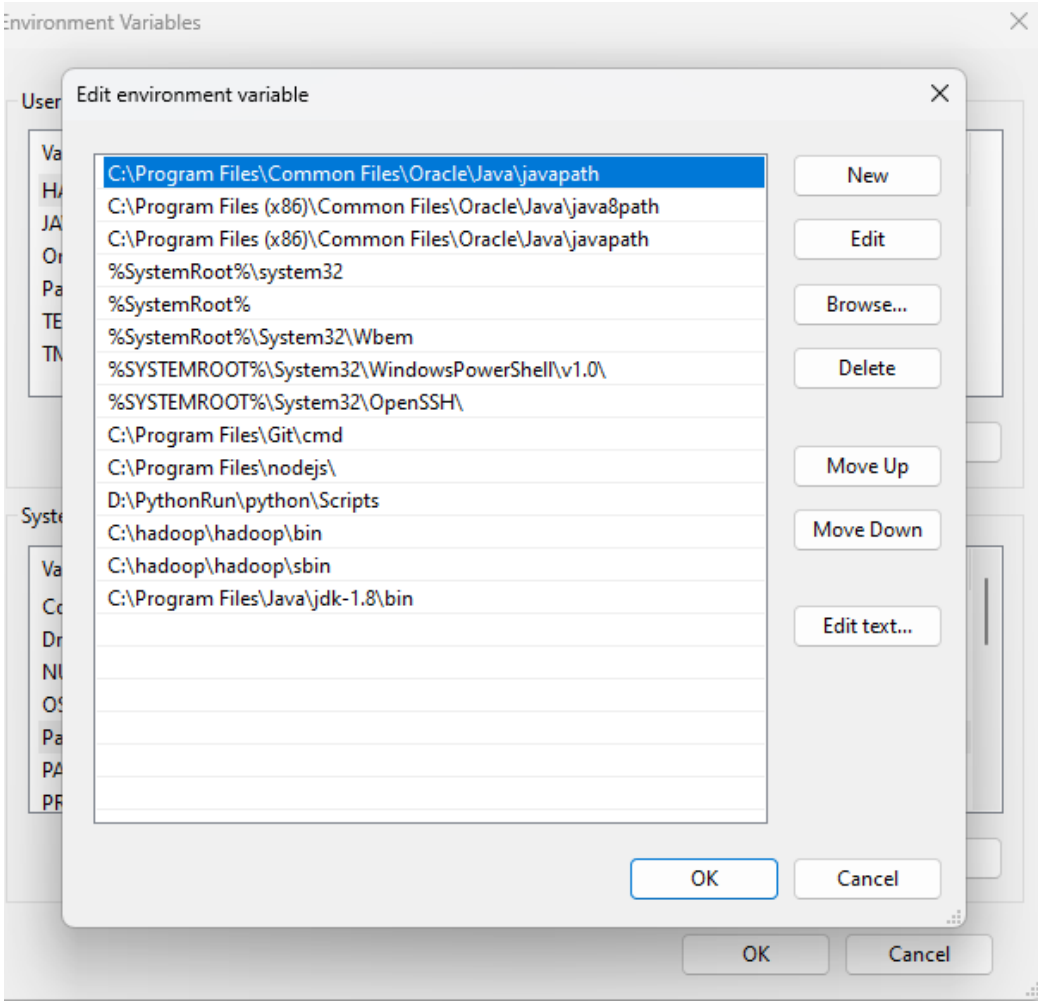
This command will start all the required Hadoop services, including the NameNode, DataNode, and JobTracker. Wait for a few minutes until all the services are started.

Installation

To ensure that Hadoop is properly installed, open a web browser and go to http://localhost:9870. This will launch the web interface for the Hadoop NameNode. You should see a page with Hadoop cluster information.

Installation Screenshots:






```
C:\Users\hp>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/hadoop/share/hadoop/common/hadoop-common-3.3.6.jar
```

```
java version "1.8.0_421"
Java(TM) SE Runtime Environment (build 1.8.0_421-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.421-b09, mixed mode)
```

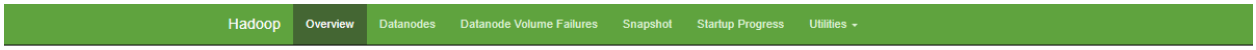
```
C:\Users\hp>hdfs namenode -format
2024-08-16 15:07:07,538 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = DESKTOP-94HDGKA/192.168.0.102
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.3.6
STARTUP_MSG:   classpath = C:\hadoop\hadoop\etc\hadoop;C:\hadoop\hadoop\share\hadoop\common;C:\hadoop\hadoop\share\hadoo
p\common\lib\animal-sniffer-annotations-1.17.jar;C:\hadoop\hadoop\share\hadoop\common\lib\audience-annotations-0.5.0.jar
;C:\hadoop\hadoop\share\hadoop\common\lib\avro-1.7.7.jar;C:\hadoop\hadoop\share\hadoop\common\lib\checker-qual-2.5.2.jar
;C:\hadoop\hadoop\share\hadoop\common\lib\commons-beanutils-1.9.4.jar;C:\hadoop\hadoop\share\hadoop\common\lib\commons-c
```

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

C:\Users\hp>jps
10696 DataNode
17512 NameNode
16620 Jps
```

```
C:\Users\hp>start-yarn.cmd
starting yarn daemons

C:\Users\hp>jps
10696 DataNode
17512 NameNode
20456 Jps
264 NodeManager
9404 ResourceManager
```



Overview 'localhost:9000' (✔active)

Started:	Fri Aug 16 15:08:58 +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-e4a3dea3-7721-4ab1-ab86-488205d901c7
Block Pool ID:	BP-160749255-192.168.0.102-1723778266263

Summary

Security is off.

Safemode is off.

7 files and directories, 3 blocks (3 replicated blocks, 0 erasure coded block groups) = 10 total filesystem object(s).

Heap Memory used 71.24 MB of 154 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 52.72 MB of 54 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	231.71 GB
Configured Remote Capacity:	0 B
DFS Used:	513 B (0%)
Non DFS Used:	157.13 GB