Install Apache Hadoop on Windows 10

Author: Sri Adilakshmi M

Table of Contents

This document outlines the steps needed to install Apache Hadoop on Windows Operating system.

1. Overview:

Apache Hadoop was introduced to handle Big Data in a distributed manner with parallel computation. Hadoop follows Master-Slave architecture in which Master node communicates to Slave nodes. Hadoop ecosystem consists of **HDFS** (Hadoop Distributed File System), **Resource Manager** (YARN) and **Computation Engine** (MapReduce).

The core components of Hadoop include `NameNode`, `DataNode`, `ResourceManager` (including `Scheduler` and `ApplicationManager`), `NodeManager` and `ApplicationMaster`.

This document provides instructions to install Hadoop 3.3.6 release by taking the reference of Hadi Fadlallah, Installing Hadoop 3.2.1 Single node cluster on Windows 10 article.

2. Prerequisites:

The following prerequisites need to be installed before running Hadoop.

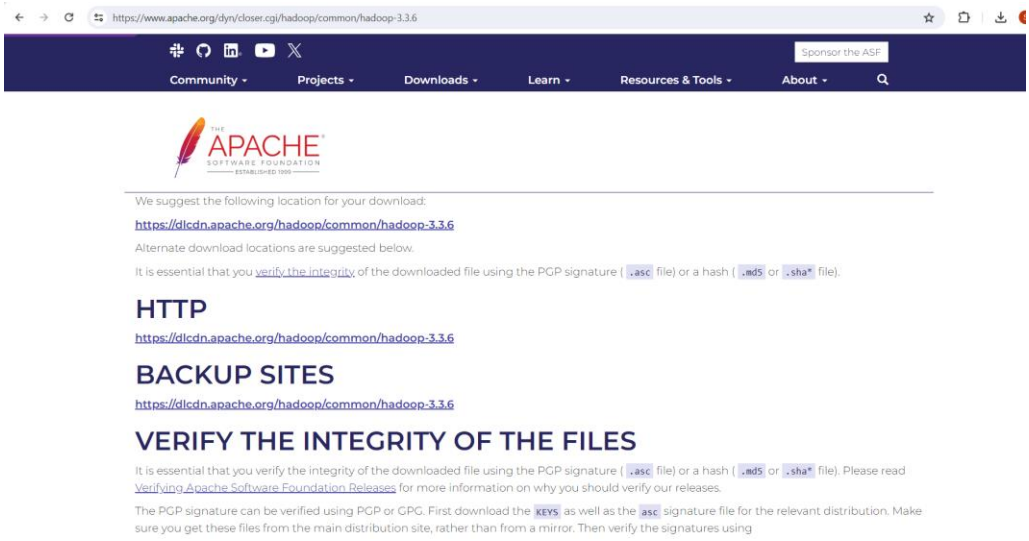
1. **JRE 8:** Hadoop 3.x requires Java 8 runtime environment *(Hadoop 3.3 supports Java 8 and Java 11 while Hadoop 3.0.x to 3.2.x supports Java8 only and Hadoop 2.7.x to 2.10.x support both Java 7 and Java 8).* See Hadoop Java versions for more details.

   We can either download just JRE 8 (Java Runtime Environment) for Windows offline installation from the official Java Download for Windows Offline website or download the whole JDK 8 (Java Development Kit) directly from Oracle Java Downloads website. For the complete JDK installation steps, look at here.

2. **File Archiver:** Any file archiver such as **7zip** or **WinRAR** is needed to unzip the downloaded Hadoop binaries. 7zip can be downloaded from the 7zip Downloads website and WinRAR can be downloaded from the RAR lab Downloads website.

3. Download Hadoop Binaries:

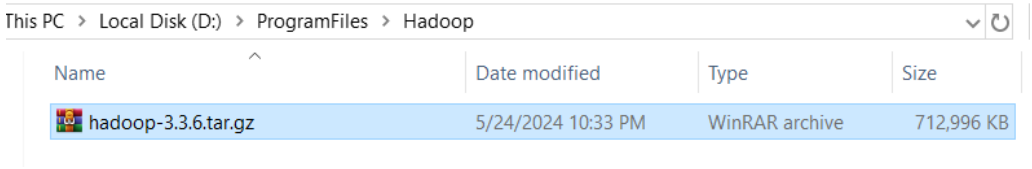
To run Hadoop, download the latest Hadoop 3.3.6 release from the Apache Hadoop Downloads mirror website.



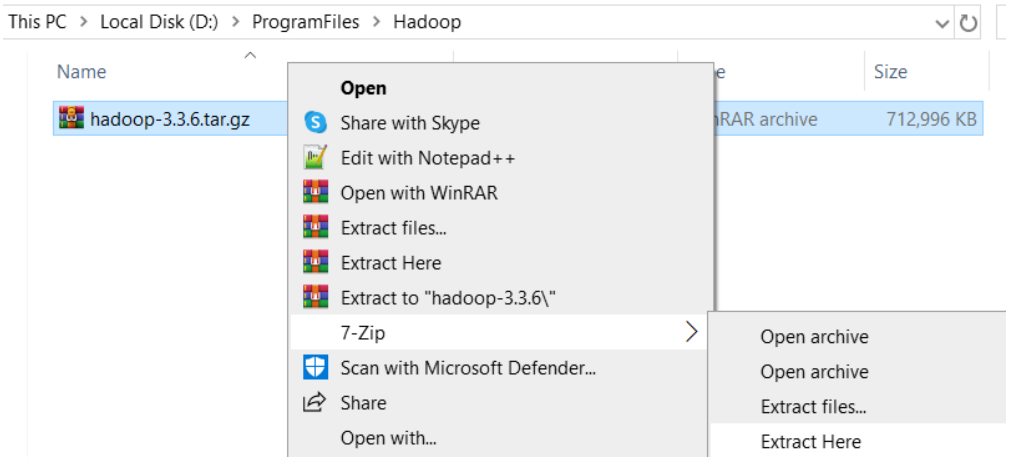
You need to download the binary file named `hadoop-3.3.6.tar.gz` file which gets downloaded to your **Downloads** folder.

After the binary `file` is downloaded:

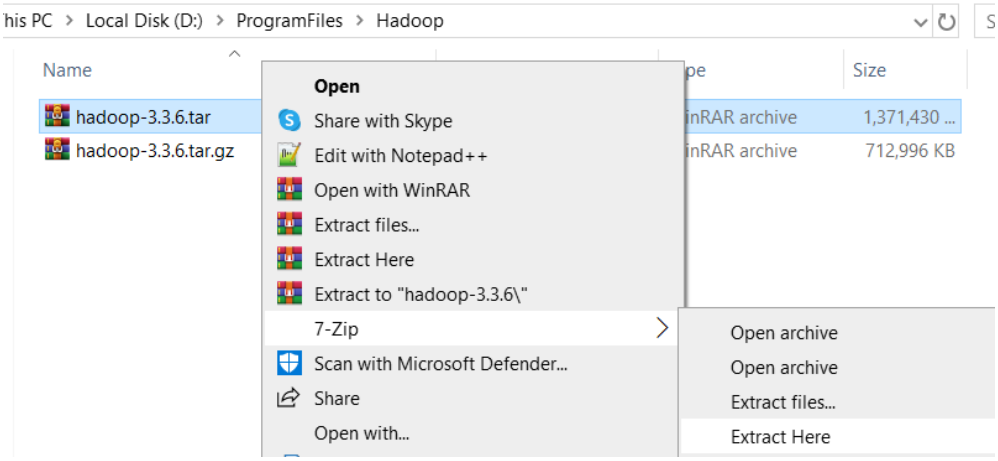
- Choose the installation directory in your machine and copy `hadoop-3.3.6.tar.gz` file to that directory. For example, I am choosing my Hadoop installation directory as `D:\ProgramFiles\Hadoop`.



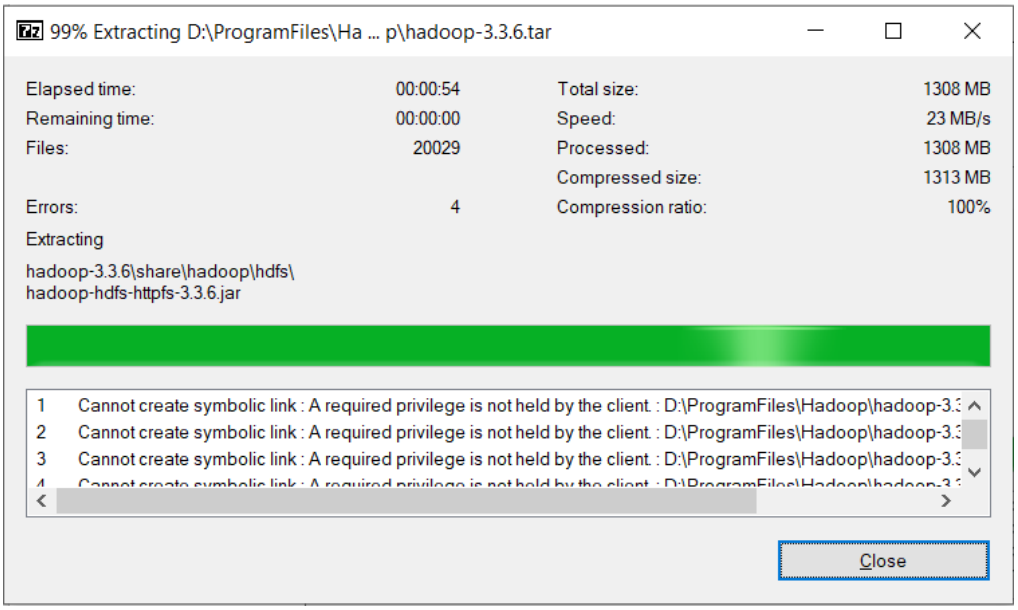
- Right click on the file and choose **7-Zip -> Extract Here** option which extracts a new packed file `hadoop-3.3.6.tar`.



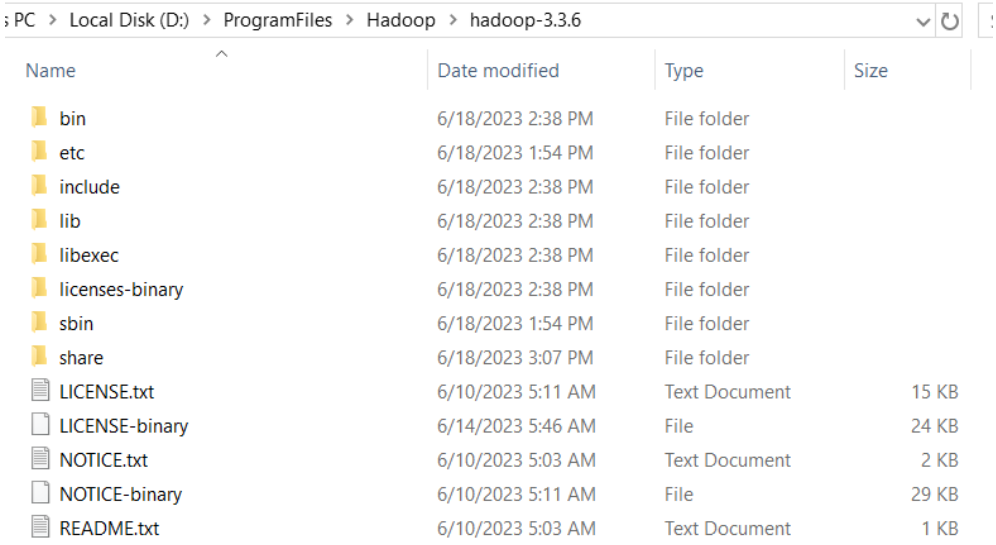
- Next, unpack `hadoop-3.3.6.tar` file using 7zip utility.



The tar file extraction may take few minutes to finish. At the end, we may see some warnings about symbolic links creation. We can ignore these warnings since they are not related to Windows operating system.



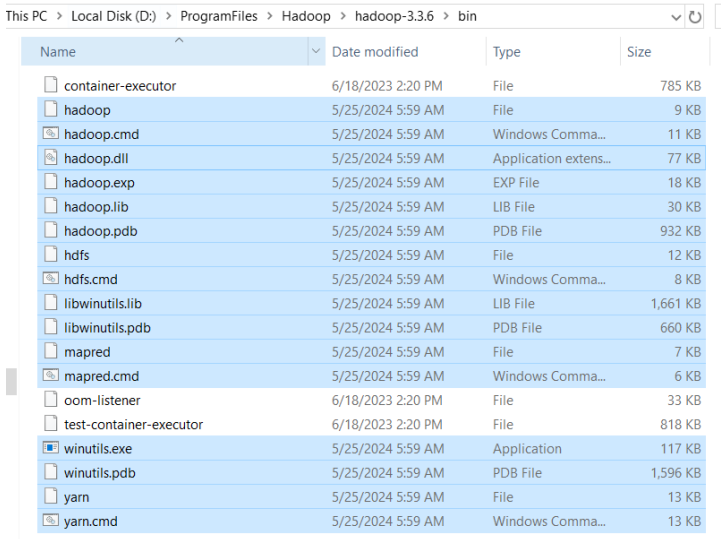
- After the file extraction is completed, you see a folder named `hadoop-3.3.6` which consists of Hadoop binaries and libraries.



Note:

Hadoop by default does not provide native IO libraries to run on Windows operating system, so we need to add Hadoop windows utilities that can be found in [cdarlint Winutils GitHub repository] for the corresponding Hadoop version installed.

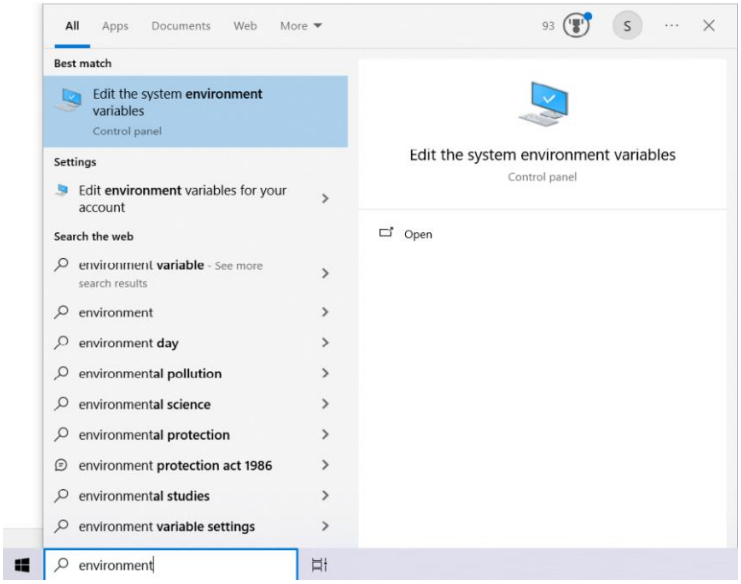
Since we installed hadoop-3.3.6 version, we need to download windows utilities for Hadoop 3.3.6 version from [hadoop-3.3.6 winutils] Github link and copy them into `hadoop-3.3.6\bin` directory (replace files if already available).



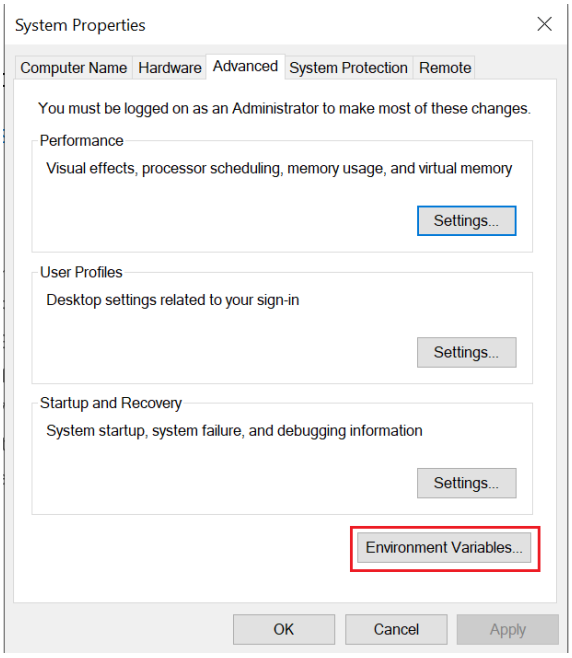
4. Set up Environment Variables:

After installing pre-requisites and Hadoop, we should configure the environment variables defining Hadoop and Java default paths.

In the Windows search bar, start typing “environment variables” and select the first match which opens up **System Properties** dialog.



On the **System Properties** window, press **Environment Variables** button.



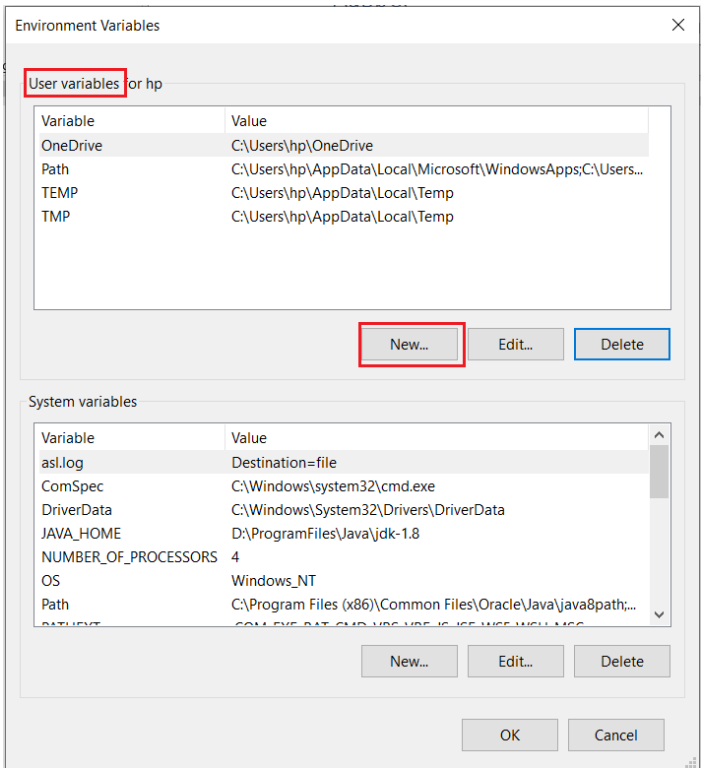
We need to setup two environment variables:

1. **JAVA_HOME**: This is the JDK installation directory path in the machine *(in my machine, it is `D:\ProgramFiles\Java\jdk-1.8`)*. Ignore it if this is already done.
2. **HADOOP_HOME**: This is the Hadoop installation directory path in the machine *(in my machine, it is `D:\ProgramFiles\Hadoop\hadoop-3.3.6`)*

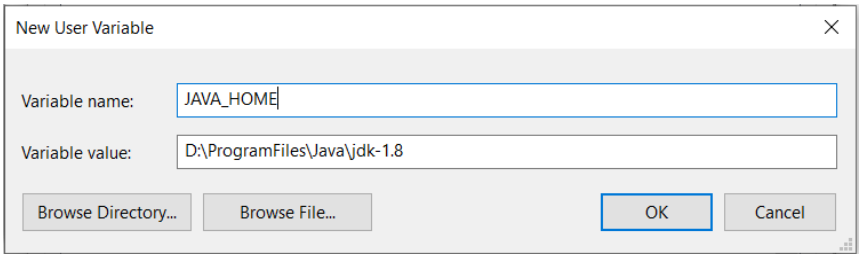
These variables need to be added to either **User environment variables** or **System environment variables** depending on Hadoop configuration needed **for a single user** or **for multiple users**.

Here, we will add User environment variables since we are configuring Hadoop for a single user. If you would like to configure Hadoop for multiple users, then define System environment variables.

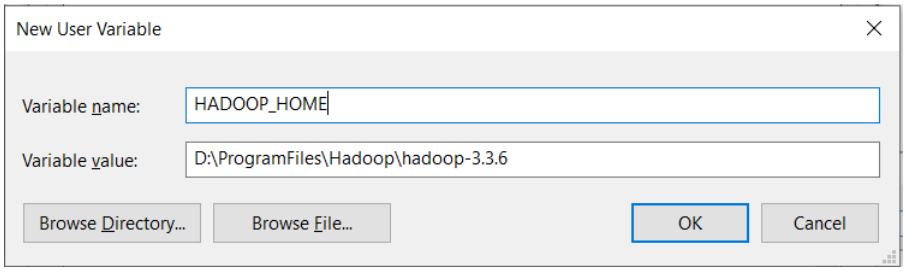
In the **Environment Variables** dialog, click on **New** under **User variables** section



Add `JAVA_HOME` variable and press OK.

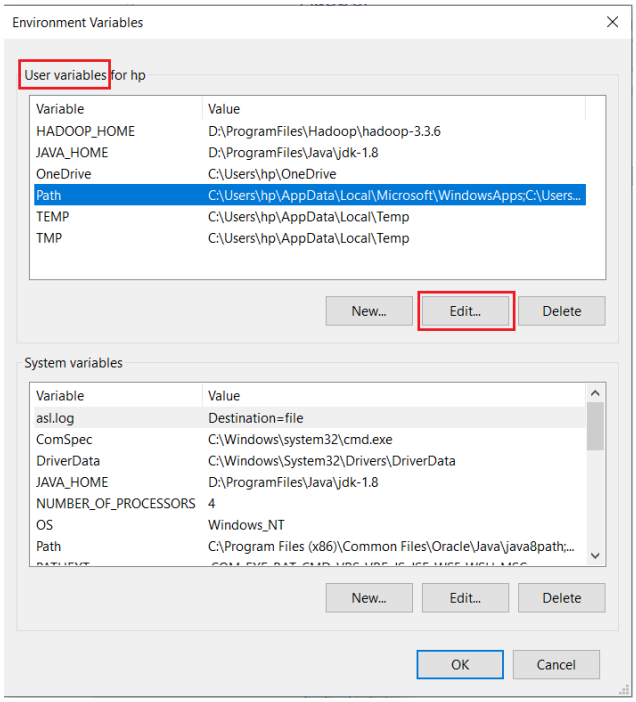


Click on **New** again and add `HADOOP_HOME` variable and press OK.

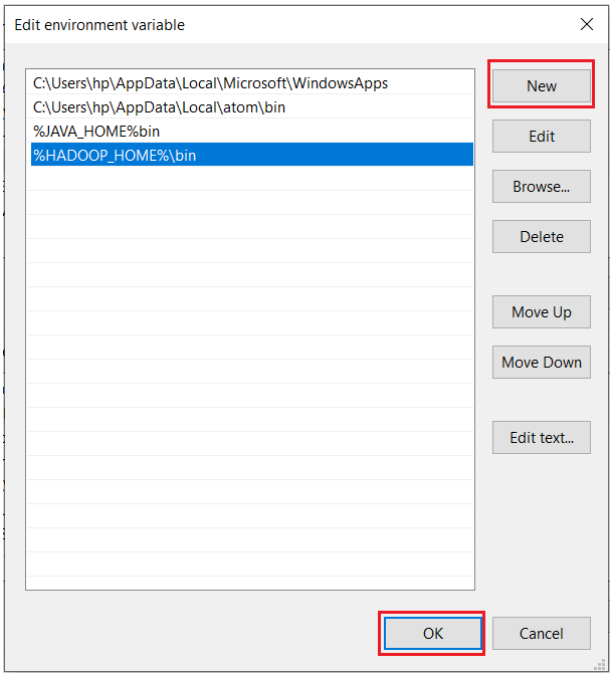


Now, we will update `PATH` variable to add Java and Hadoop binary paths.

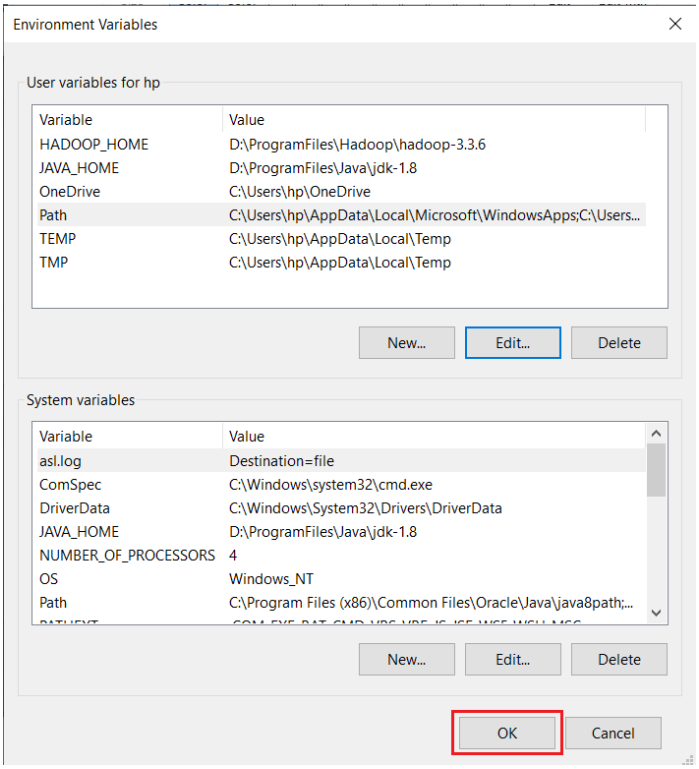
Select `PATH` variable under **User variables** and press **Edit** button.



Press **New** and add `%JAVA_HOME%\bin` and `%HADOOP_HOME%\bin` values and press OK.



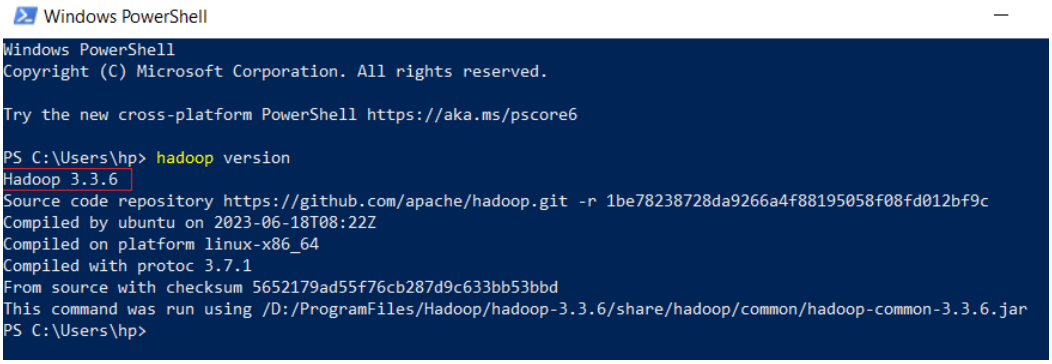
Press OK to apply environment variable changes and close window.



5. Verify Hadoop Installation:

Open **Windows PowerShell** or **Command Prompt** and verify if Hadoop is installed properly by running the following command:

```
hadoop version
```



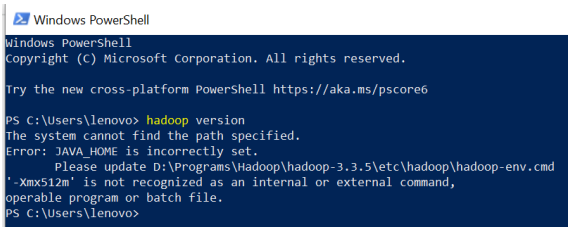
It shows **Hadoop 3.3.6** version which indicates that Hadoop has been installed successfully.

5.1. Common Errors:

1. **JAVA_HOME incorrectly set Error:**

   During the validation of hadoop installation, we might get error the following error:

   JAVA_HOME is incorrectly set



   This error generally occurs when there is a space in the JAVA_HOME path where Java is installed in the default location `"C:\Program Files\Java" or "C:\Program Files (x86)\Java"`.

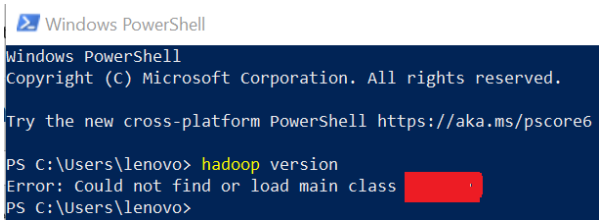
   To resolve this issue, use windows 8.3 path in the JAVA_HOME variable. As an example:

   - Use `Progra~1` instead of `Program Files`
   - Use `Progra~2` instead of `Program Files (x86)`

2. **Could not load main class Error:**

   During the validation of hadoop installation, we might get error the following error:

   Error: Could not find or load main class



   This error generally occurs when Hadoop takes our PC name as the default username, which generally contains spaces, which is not allowed.

   To resolve this issue, go to `HADOOP_HOME\etc\hadoop` location and open `hadoop-env.cmd` file with any editor such as Notepad++ and then at the last line, replace `%USERNAME%` with our name without blankspaces.

   For example:

```
set HADOOP_IDENT_STRING=SriLakshmi
```

6. Configure Hadoop Cluster:

After Hadoop has been installed, we need to modify the following four files to configure the Hadoop cluster:

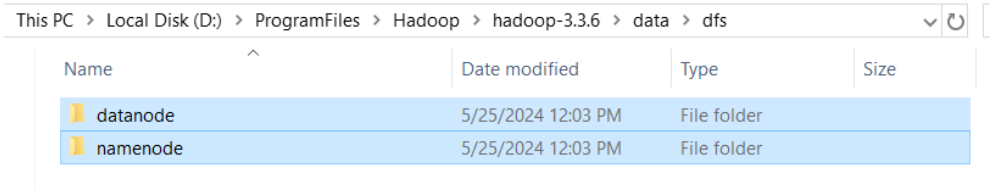
```
HADOOP_HOME\etc\hadoop\hdfs-site.xml
HADOOP_HOME\etc\hadoop\core-site.xml
HADOOP_HOME\etc\hadoop\mapred-site.xml
HADOOP_HOME\etc\hadoop\yarn-site.xml
```

6.1. HDFS Site Configuration:

As we know, Hadoop is built on Master-Slave architecture, we should first create a directory to store all master (Namenode) data and another directory to store other data (Datanode) before modifying HDFS configuraiton file.

Go to `HADOOP_HOME` directory and create a `data` folder in which create `dfs` folder. Inside `dfs` folder, create `namenode` and `datanode` subfolders.
Since we instaled Hadoop in `D:\ProgramFiles\Hadoop\hadoop-3.3.6` location, the directory structure would look like below

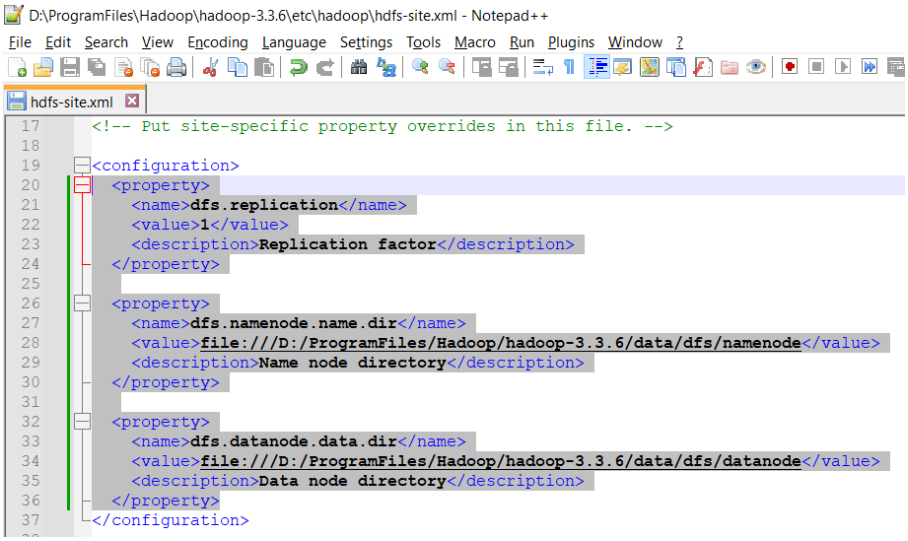
```
D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\namenode
D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode
```



Next, open `hdfs-site.xml` file located in `HADOOP_HOME\etc\hadoop` directory, and add the following properties within the `<configuration></configuration>` element. Make sure that namenode directory and datanode directory paths are valid.

```
  <property>
        <name>dfs.replication</name>
        <value>1</value>
        <description>Replication factor</description>
  </property>

  <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:///D:/ProgramFiles/Hadoop/hadoop-
3.3.6/data/dfs/namenode</value>
        <description>Name node directory</description>
  </property>
```

```
  <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:///D:/ProgramFiles/Hadoop/hadoop-
3.3.6/data/dfs/datanode</value>
        <description>Data node directory</description>
  </property>
```



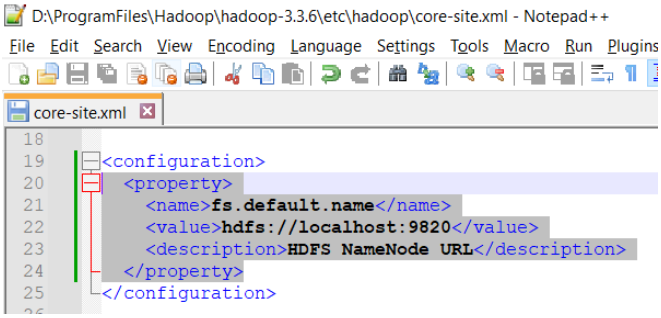
Note: *We have set the replication factor to 1 since we are creating a single node cluster.*

6.2. Core Site Configuration:

Now, we need to configure the NameNode URL which is `hdfs://localhost:9820`.

Open `core-site.xml` file located in `HADOOP_HOME\etc\hadoop` directory, and add the following properties within the `<configuration></configuration>` element:

```
  <property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9820</value>
        <description>HDFS NameNode URL</description>
  </property>
```

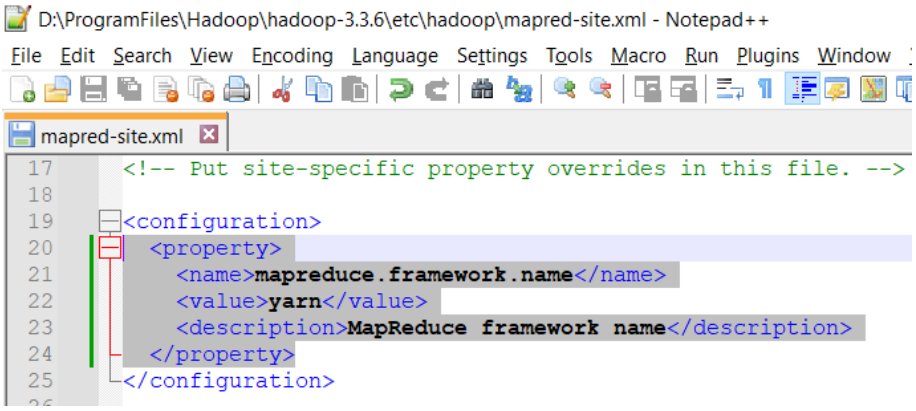


6.3. MapReduce Site Configuration:

Now, we should configure the MapReduce framework.

Open `mapred-site.xml` file located in `HADOOP_HOME\etc\hadoop` directory, and add the following properties within the `<configuration></configuration>` element:

```
<property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
      <description>MapReduce framework name</description>
</property>
```

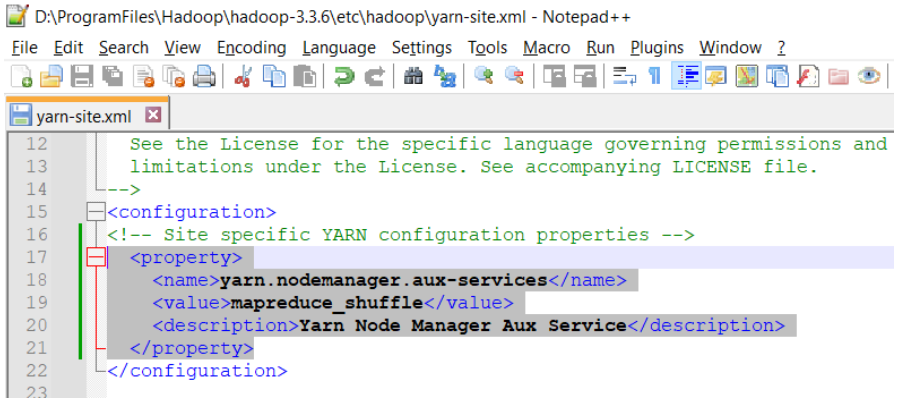


6.4. YARN Site Configuration:

Now, we should configure the YARN site.

Open `yarn-site.xml` file located in `HADOOP_HOME\etc\hadoop` directory, and add the following XML code within the `<configuration></configuration>` element:

```
<property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
      <description>Yarn Node Manager Aux Service</description>
</property>
```

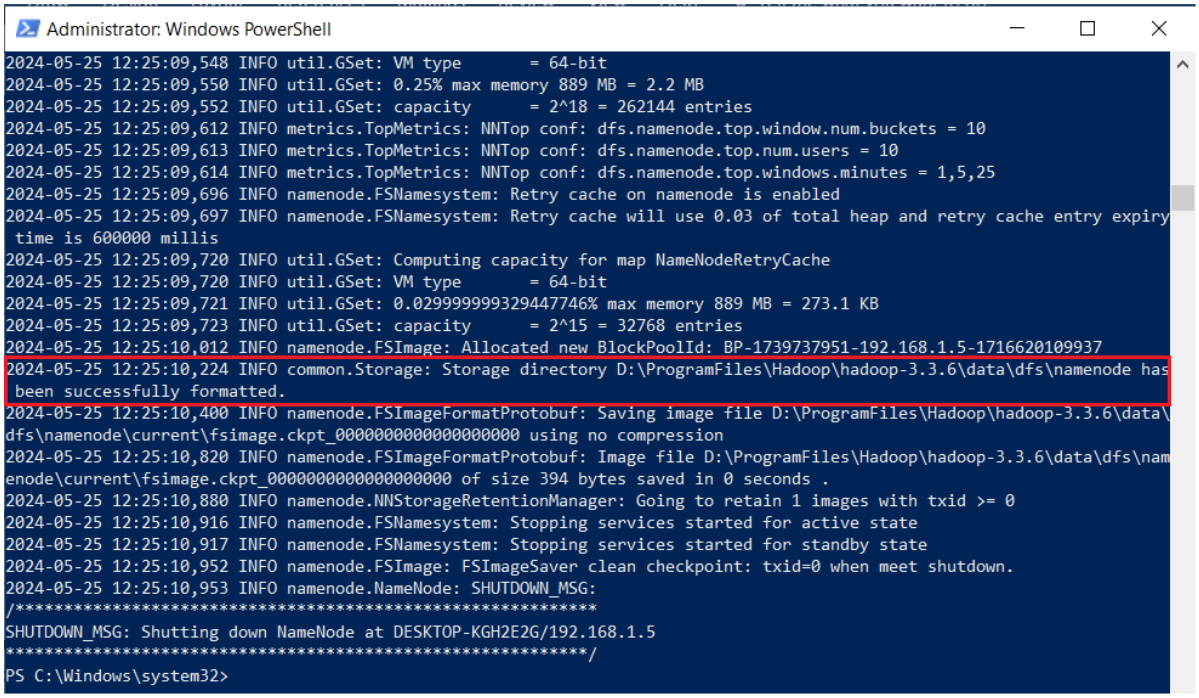


7. Format NameNode:

After completing the Hadoop configuration, we need to format the NameNode to bring the above configuration changes into effect.

Open **Windows PowerShell** in **Administrator** mode and execute this command:

```
hdfs namenode -format
```



It shows us a message that Storage directory has been successfully formatted.

8. Start Hadoop Services:

Now, open **Windows PowerShell** or **Command Prompt** as **Administrator** and navigate to `HADOOP_HOME\sbin` directory to start services

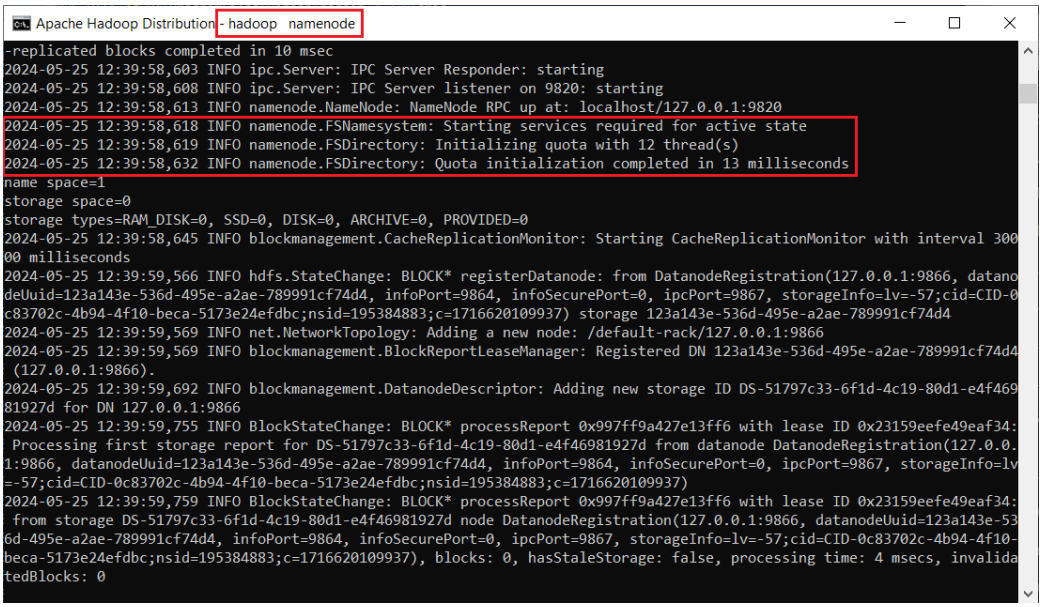
8.1. Start Hadoop Nodes:

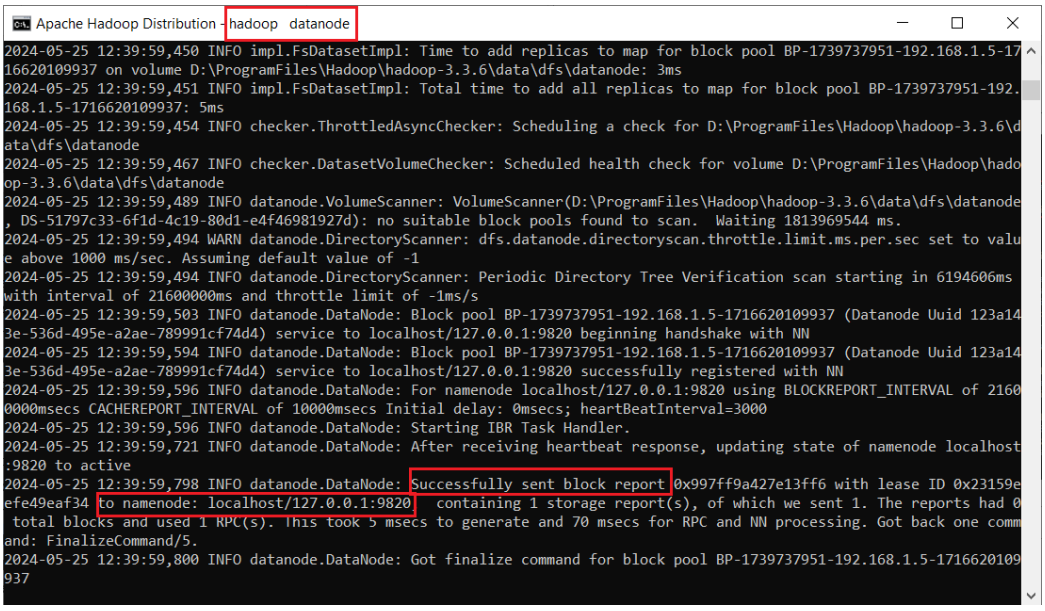
Run the following command to start the Hadoop nodes.

```
cd D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin
.\start-dfs.cmd
```

```
Windows PowerShell
PS C:\Users\hp> d:
PS D:\> cd D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin
PS D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin> .\start-dfs.cmd
PS D:\ProgramFiles\Hadoop\hadoop-3.3.6\sbin>
```

After executing the above command, it opens up two command prompt windows - one for the **namenode** and other for the **datanode** as below. Wait until `namenode` service says *"Quota initialization completed"* and `datanode` service says *"Successfully sent block report to namenode: localhost/127.0.0.1:9820"*.

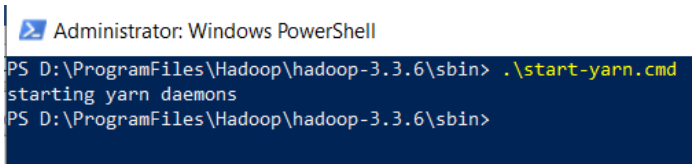




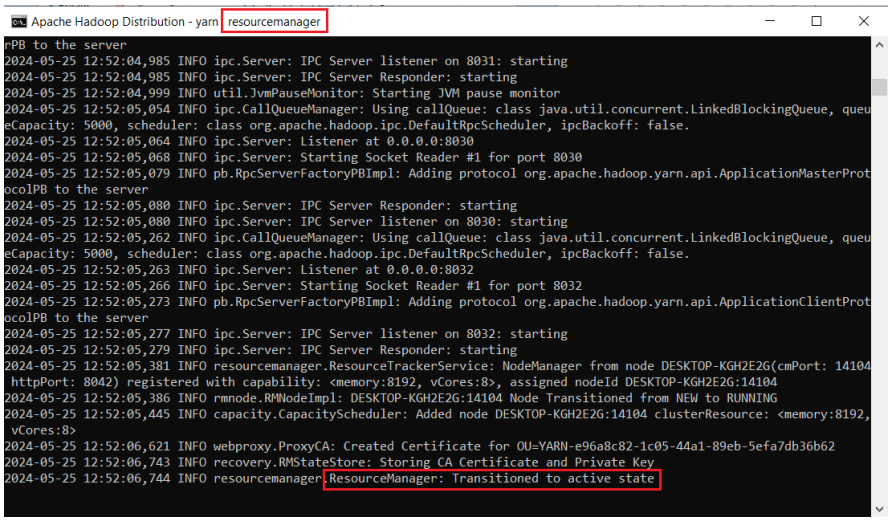
8.2. Start Hadoop YARN:

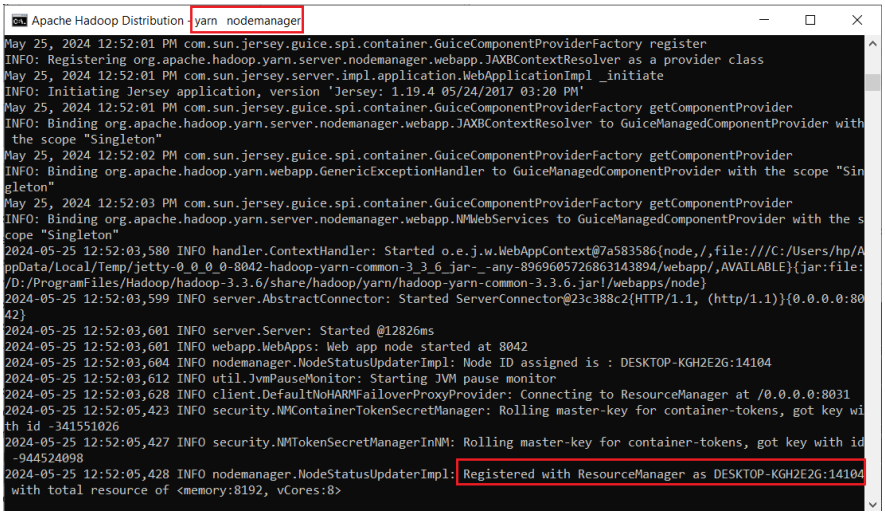
Next, start the Hadoop YARN services using the following command

```
.\start-yarn.cmd
```



After executing the above command, it opens up two command prompt windows - one for the **resourcemanager** and other for the **nodemanager** as below. Wait until `resourcemanager` service says "*Transitioned to active state*" and `nodemanager` service says "*Registered with ResourceManager*"





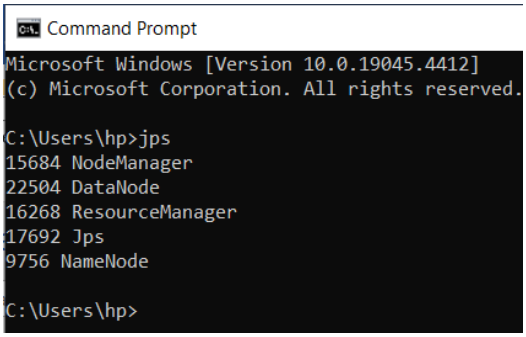
8.3. Verify Services:

Now, run the following command to make sure all services are started successfully:

```
jps
```

It should display the following services:

```
15684 NodeManager
22504 DataNode
16268 ResourceManager
17692 Jps
9756 NameNode
```



9. Run HDFS Commands:

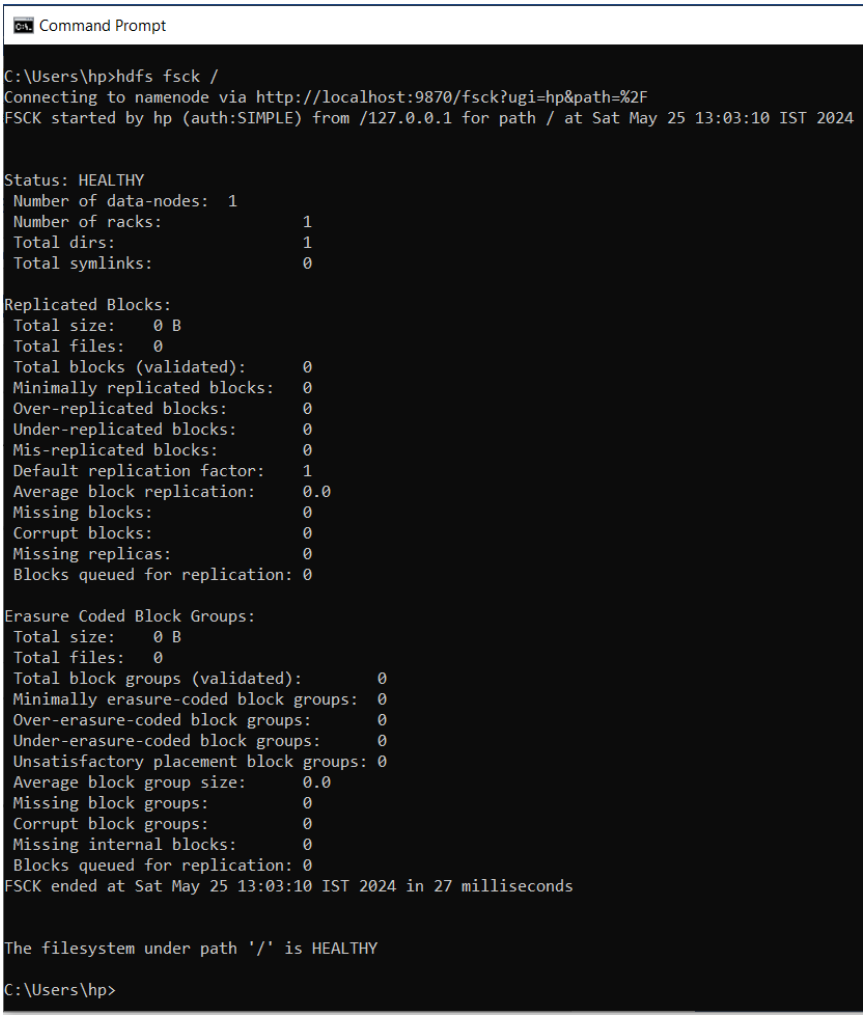
Let us run a few `hdfs` commands to verify if they are working without any issue.

9.1. Verify File System:

Check the status of Hadoop file system by running the following command.

```
hdfs fsck /
```

It should display a message *"The filesystem under path '/' is HEALTHY"* indicating that Hadoop root file system (identified by **/**) does not have any corrupted or missing data blocks.



9.2. List Contents:

Run the following command to list all contents of the root directory (`'/'`)

```
hadoop fs -ls /
```

9.3. Create Directory:

Run the following command to create a directory named `user` under root directory (`'/'`)

```
hadoop fs -mkdir /user
```

It should create a directory in HDFS file system.

```
C:\Users\hp>hadoop fs -ls /

C:\Users\hp>hadoop fs -mkdir /user

C:\Users\hp>hadoop fs -ls /
Found 1 items
drwxr-xr-x   - hp supergroup          0 2024-05-25 13:06 /user

C:\Users\hp>
```

9.4. Copy File:

Run this command to copy a file named `sample_file.txt` into HDFS at `/user` path.

```
hadoop fs -copyFromLocal sample_file.txt /user
```

It should copy the given file into HDFS at `/user` path

```
C:\Users\hp>hadoop fs -copyFromLocal dblook_emp_ddl.sql /user

C:\Users\hp>hadoop fs -ls /user
Found 1 items
-rw-r--r--   1 hp supergroup        326 2024-05-25 13:09 /user/dblook_emp_ddl.sql

C:\Users\hp>
```

9.5. Remove File:

Remove the file from HDFS using the following command

```
hadoop fs -rm /user/<file_name>
```

It should remove the given file from HDFS.

```
C:\Users\hp>hadoop fs -rm /user/dblook_emp_ddl.sql
Deleted /user/dblook_emp_ddl.sql

C:\Users\hp>hadoop fs -ls /user

C:\Users\hp>
```

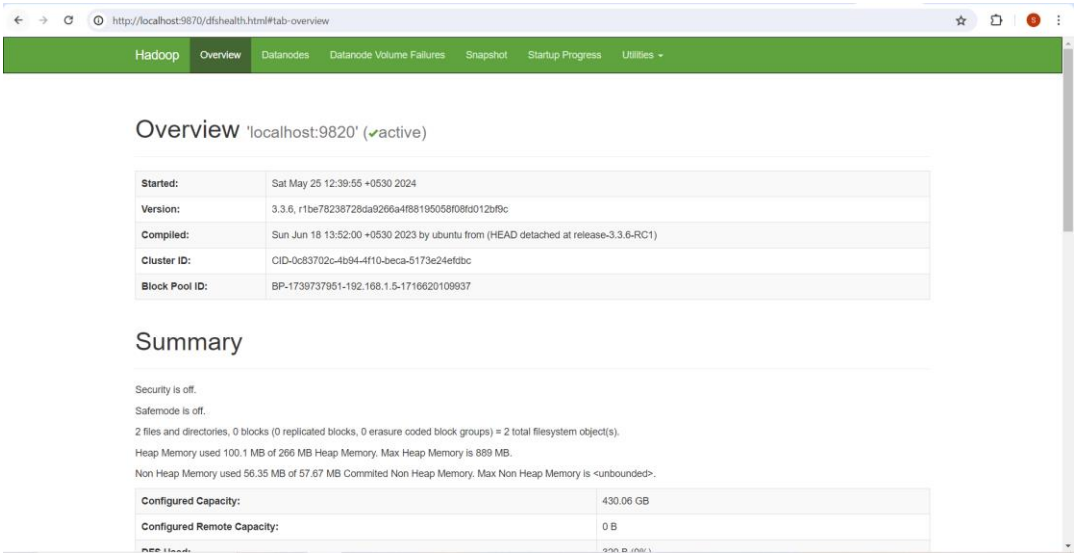
Similarly, we can run any other HDFS commands here.

10. Hadoop Web UI:

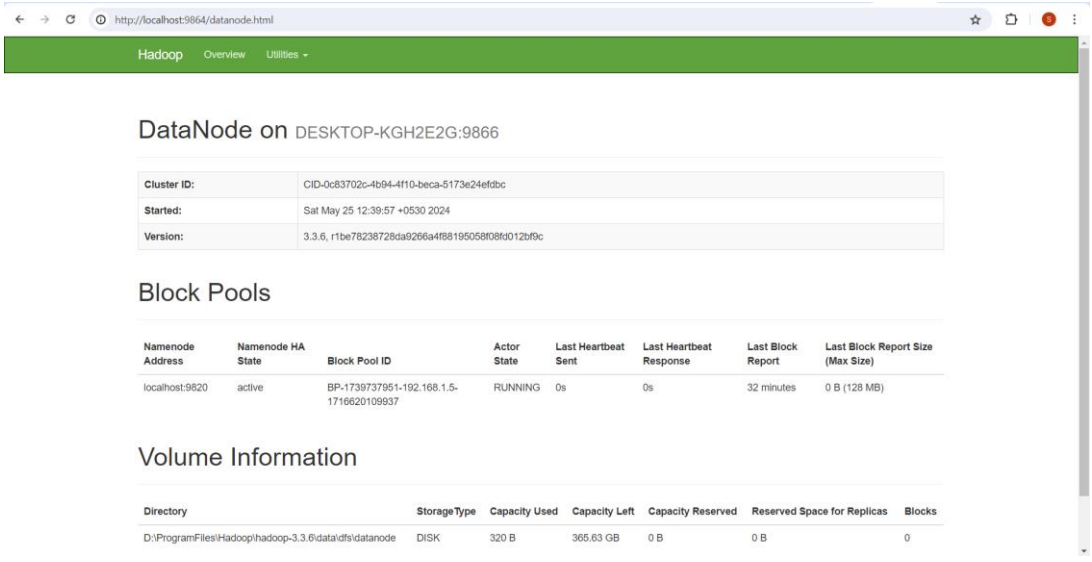
Hadoop provides three web interfaces that can be used for monitoring NameNode, DataNode and YARN resources.

- Name Node UI
- Data Node UI
- YARN UI

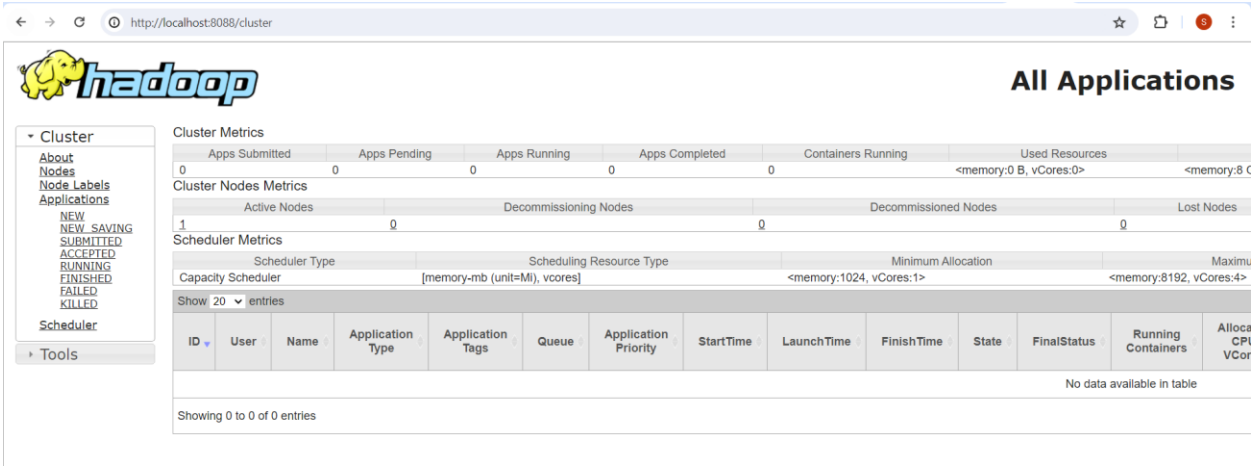
NameNode UI: http://localhost:9870/dfshealth.html



DataNode UI: http://localhost:9864/datanode.html



YARN UI: http://localhost:8088/cluster



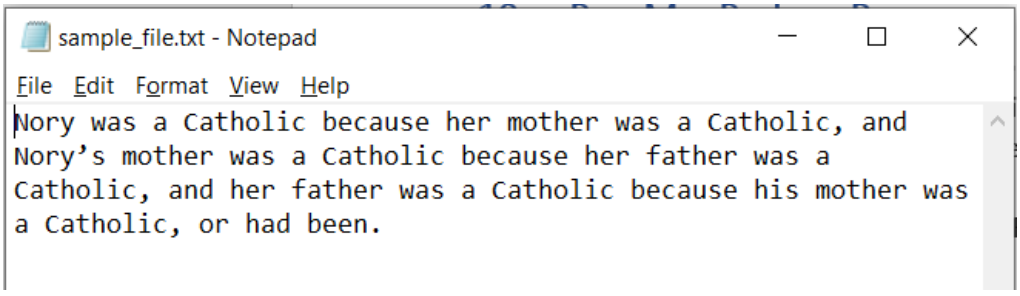
11. MapReduce Examples:

Hadoop MapReduce is a software framework built for writing applications to process huge amounts of data *(multi-terabyte data-sets)* in-parallel on large clusters *(thousands of nodes)* of commodity hardware in a reliable, fault-tolerant manner.

Hadoop 3.3.6 version provides in-built MapReduce example programs such as `wordcount, wordmean, aggregatewordcount, sudoku, sort`, etc. that can be executed on Hadoop cluster. These programs are packaged under `hadoop-mapreduce-examples-3.3.6.jar` file located at `HADOOP_HOME\share\hadoop\mapreduce` directory.

11.1. Run WordCount Program:

Let us execute the `wordcount` example which counts each word in the input file.
First, create a file named `sample_file.txt` with some random text.



Next, run the following commands to create an input directory and move the above file into HDFS

```
hadoop fs -mkdir /input
hadoop fs -put sample_file.txt /input
hadoop fs -ls /input
```

```
D:\Big Data\Datasets>hadoop fs -mkdir /input

D:\Big Data\Datasets>hadoop fs -put sample_file.txt /input

D:\Big Data\Datasets>hadoop fs -ls /input
Found 1 items
-rw-r--r--   1 hp supergroup        202 2024-05-25 13:32 /input/sample_file.txt

D:\Big Data\Datasets>
```

Now, run the `wordcount` program using the following command.

```
hadoop jar D:\ProgramFiles\Hadoop\hadoop-3.3.6\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3.6.jar wordcount /input/sample_file.txt /output/wordcount
```

```
Command Prompt

D:\Big Data\Datasets>hadoop jar D:\ProgramFiles\Hadoop\hadoop-3.3.6\share\hadoop\mapreduce\hadoop-mapreduce-examples-3.3
.6.jar wordcount /input/sample_file.txt /output/wordcount
2024-05-25 13:36:19,173 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-05-25 13:36:20,676 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/
hp/.staging/job_1716621724430_0001
2024-05-25 13:36:21,294 INFO input.FileInputFormat: Total input files to process : 1
2024-05-25 13:36:21,748 INFO mapreduce.JobSubmitter: number of splits:1
2024-05-25 13:36:22,025 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1716621724430_0001
2024-05-25 13:36:22,025 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-05-25 13:36:22,403 INFO conf.Configuration: resource-types.xml not found
2024-05-25 13:36:22,404 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-05-25 13:36:23,240 INFO impl.YarnClientImpl: Submitted application application_1716621724430_0001
2024-05-25 13:36:23,376 INFO mapreduce.Job: The url to track the job: http://DESKTOP-KGH2E2G:8088/proxy/application_1716
621724430_0001/
2024-05-25 13:36:23,378 INFO mapreduce.Job: Running job: job_1716621724430_0001
2024-05-25 13:36:37,959 INFO mapreduce.Job: Job job_1716621724430_0001 running in uber mode : false
2024-05-25 13:36:37,961 INFO mapreduce.Job:  map 0% reduce 0%
2024-05-25 13:36:46,190 INFO mapreduce.Job:  map 100% reduce 0%
2024-05-25 13:36:53,293 INFO mapreduce.Job:  map 100% reduce 100%
2024-05-25 13:36:54,326 INFO mapreduce.Job: Job job_1716621724430_0001 completed successfully
2024-05-25 13:36:54,546 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=182
                FILE: Number of bytes written=555099
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=310
                HDFS: Number of bytes written=116
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=5267
                Total time spent by all reduces in occupied slots (ms)=5223
                Total time spent by all map tasks (ms)=5267
                Total time spent by all reduce tasks (ms)=5223
                Total vcore-milliseconds taken by all map tasks=5267
                Total vcore-milliseconds taken by all reduce tasks=5223
                Total megabyte-milliseconds taken by all map tasks=5393408
                Total megabyte-milliseconds taken by all reduce tasks=5348352
```

11.2. Validate Output in HDFS:

Let us verify the output generated by `wordcount` program in HDFS by running the following commands

```
hadoop fs -ls /output/wordcount
hadoop fs -cat /output/wordcount/part-r-00000
```

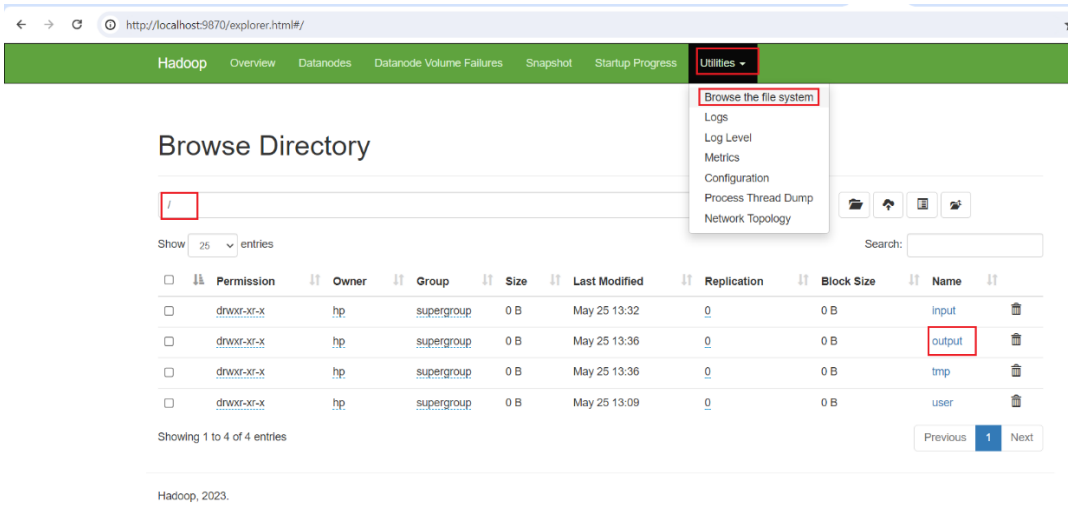
```
D:\Big Data\Datasets>hadoop fs -ls /output/wordcount
Found 2 items
-rw-r--r--   1 hp supergroup          0 2024-05-25 13:36 /output/wordcount/_SUCCESS
-rw-r--r--   1 hp supergroup        116 2024-05-25 13:36 /output/wordcount/part-r-00000

D:\Big Data\Datasets>hadoop fs -cat /output/wordcount/part-r-00000
Catholic        3
Catholic,       3
Nory    1
NoryΓÇÖs        1
a       6
and     2
because 3
been.   1
father  2
had     1
her     3
his     1
mother  3
or      1
was     6

D:\Big Data\Datasets>
```

The program has generated the output file `part-r-00000` in HDFS under `/output/wordcount/` directory and the file contains count of each words from the input file.

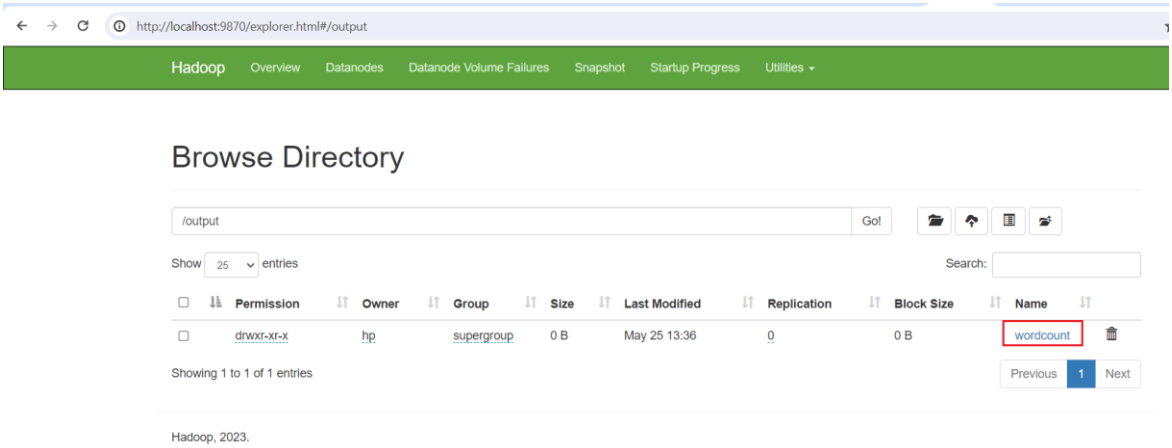
11.3. Review NameNode UI:

The above output is visible in NameNode UI http://localhost:9870/dfshealth.html as well.

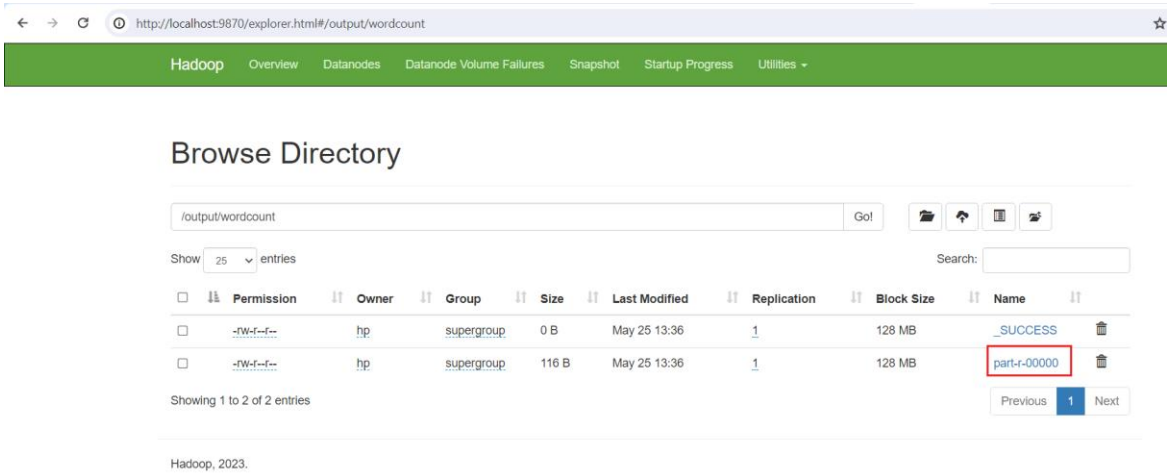
In NameNode UI, go to **Utilities** tab and select **Browse the file system** which displays the list of folders and files created under the root (/) directory in HDFS where click on `output` folder.



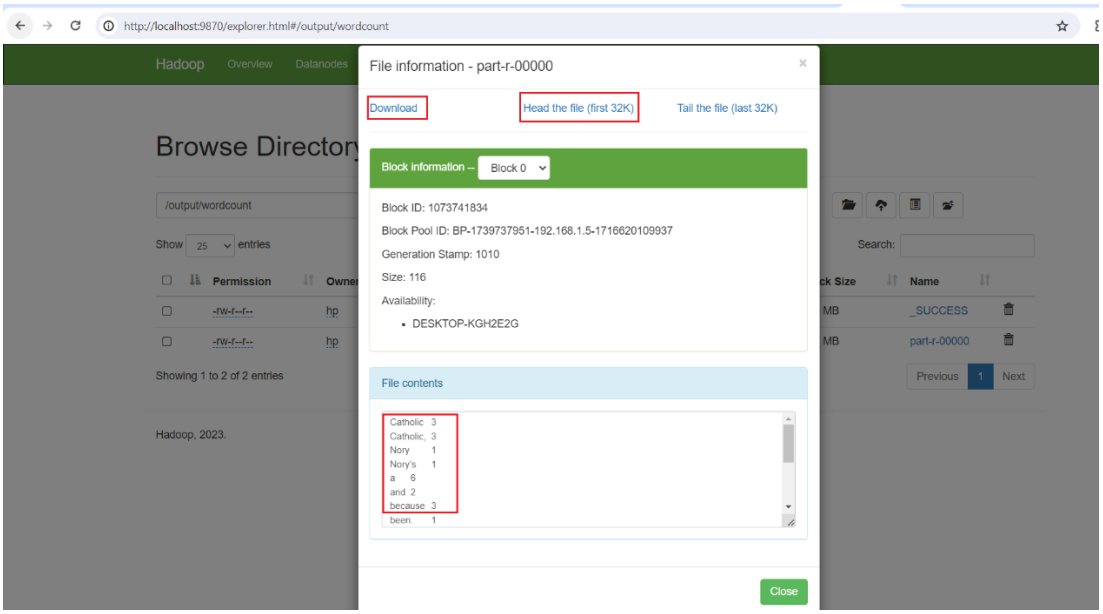
Then, click on `wordcount` folder.



Then click on `part-r-00000` file.

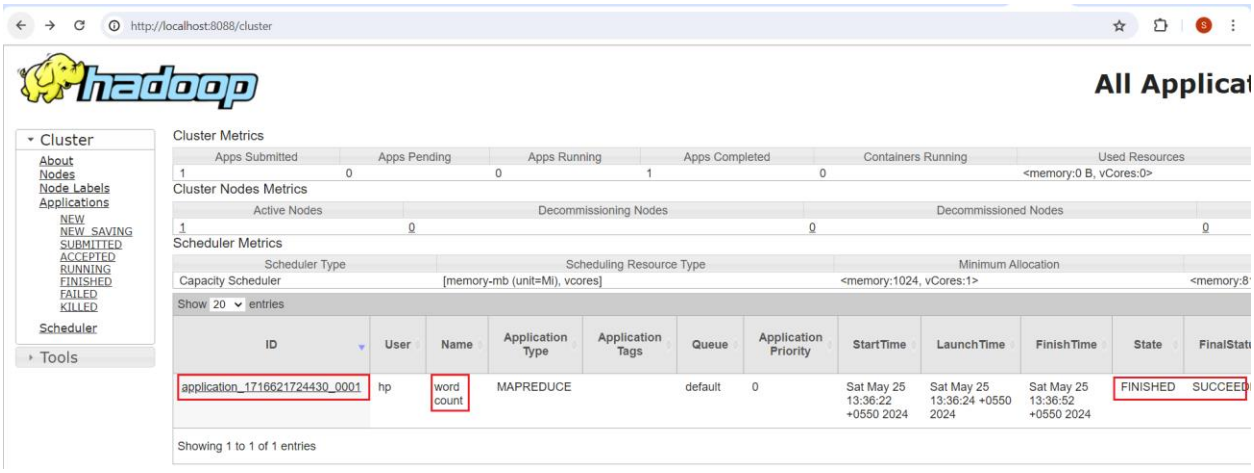


Then, we can see the file information such as Block Id, Block Pool ID, Generation stamp etc. Click on **Head the file** tab where we can see the first few lines in the `part-r-00000` file. We can also **Download** this file into our local system.

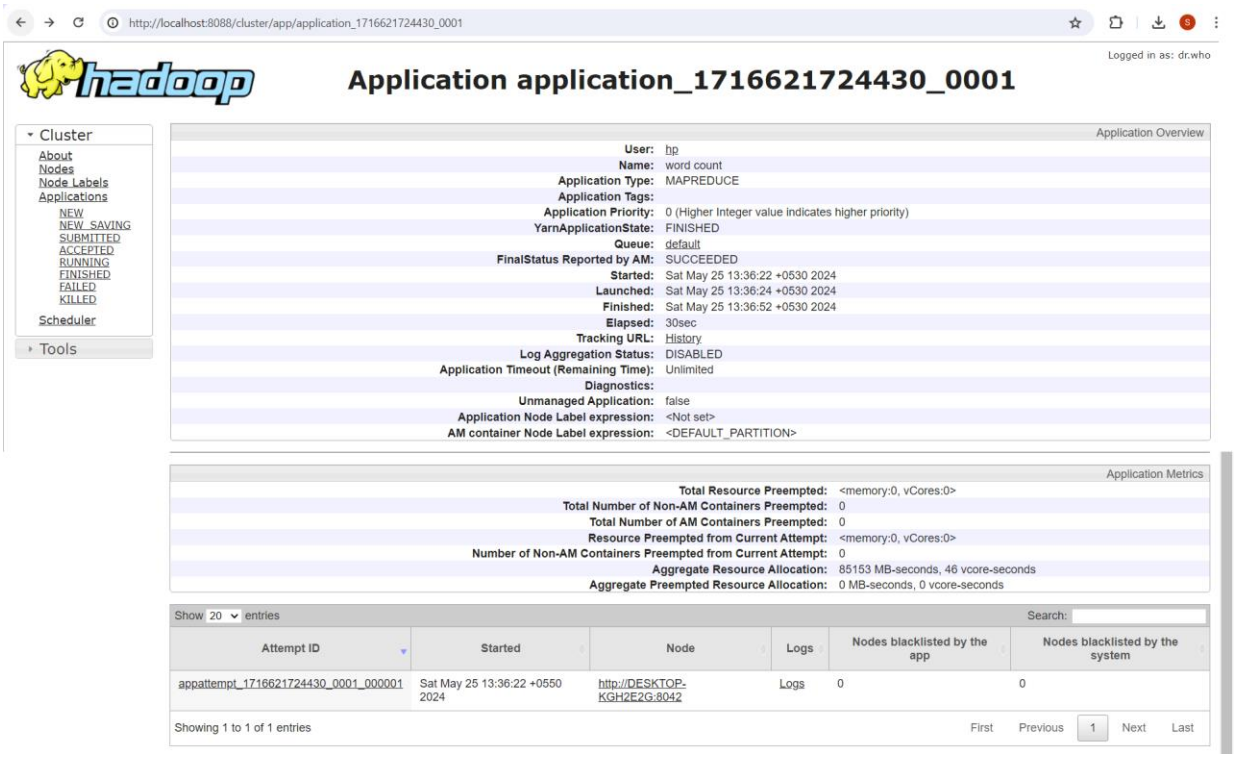


11.4. Review Job Details in YARN UI:

Open YARN UI http://localhost:8088/cluster where we can see the MapReduce job with application name `wordcount` that was executed and finished successfully.



Click on the application ID above to see the complete details of the application as shown below.



With this, we can say that our Hadoop 3.3.6 version has been installed and is working successfully.