

Install Apache Hadoop 3.3.6 on Windows 10

Author: Sri Adilakshmi M

Table of Contents

1. Overview:	4
2. Prerequisites:	4
3. Download Hadoop Binaries:	5
4. Set up Environment Variables:	8
5. Verify Hadoop Installation:	12
5.1. Common Errors:	13
6. Configure Hadoop Cluster:	14
6.1. HDFS Site Configuration:	14
6.2. Core Site Configuration:	15
6.3. MapReduce Site Configuration:	16
6.4. YARN Site Configuration:	16
7. Format NameNode:	17
8. Start Hadoop Services:	17
8.1. Start Hadoop Nodes:	17
8.2. Start Hadoop YARN:	19
8.3. Verify Services:	20
9. Run HDFS Commands:	20
9.1. Verify File System:	20
9.2. List Contents:	21
9.3. Create Directory:	21
9.4. Copy File:	22
9.5. Remove File:	22
10. Hadoop Web UI:	23
11. MapReduce Examples:	24
11.1. Run WordCount Program:	24
11.2. Validate Output in HDFS:	26
11.3. Review NameNode UI:	26
11.4. Review Job Details in YARN UI:	28
12. Stop Hadoop Services:	29

12.1.	Stop Hadoop Nodes:	29
12.2.	Stop Hadoop YARN:	30
12.3.	Verify Services:	30

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.

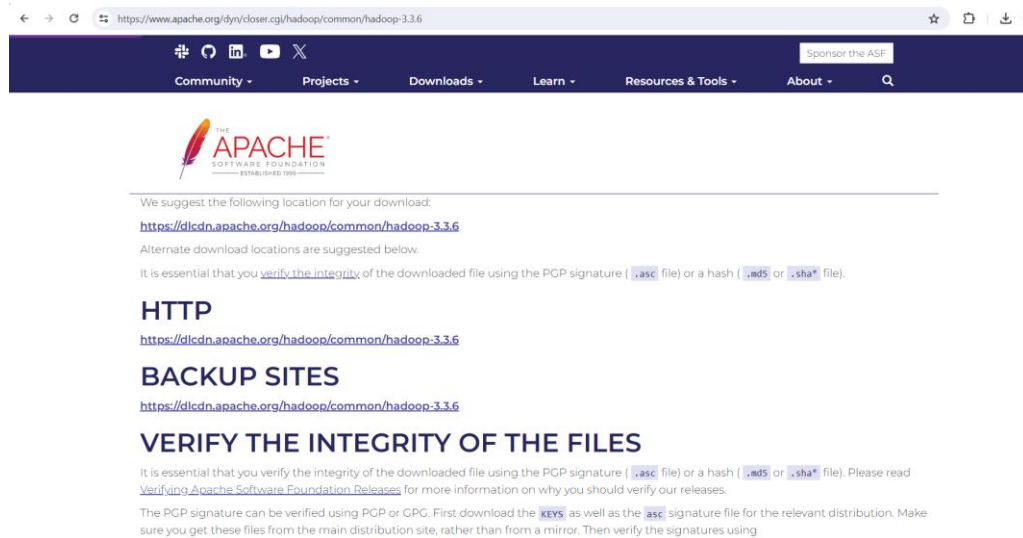
1. **JRE 8:** Hadoop 3.x requires Java 8 runtime environment (*Hadoop 3.3 supports Java 8 and Java 11 while Hadoop 3.0.x to 3.2.x supports Java8 only and Hadoop 2.7.x to 2.10.x support both Java 7 and Java 8*). See [Hadoop Java versions](#) for more details.

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](https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/) mirror website.



Go to the [suggested location](https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/) for download from where you need to download the binary file named `hadoop-3.3.6.tar.gz` file which gets downloaded to your **Downloads** folder.


A screenshot of the Apache Hadoop Downloads page for version 3.3.6. The page has a dark blue header with navigation links: Community, Projects, Downloads, Learn, Resources & Tools, and About. Below the header is the Apache Software Foundation logo. The main content area is white and contains the following text: "We suggest the following location for your download: https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6". It also lists alternate download locations and provides instructions on how to verify the integrity of the downloaded file using PGP signatures or hashes. The page is titled "Index of /hadoop/common/hadoop-3.3.6/".

Name	Last modified	Size	Description
Parent Directory	-	-	-
CHANGELOG.md	2023-06-25 23:35	22K	
CHANGELOG.md.asc	2023-06-25 23:35	833	
CHANGELOG.md.sha512	2023-06-25 23:35	153	
RELEASENOTES.md	2023-06-25 23:35	1.4K	
RELEASENOTES.md.asc	2023-06-25 23:35	833	
RELEASENOTES.md.sha512	2023-06-25 23:35	156	
hadoop-3.3.6-aarch64.tar.gz	2023-06-25 23:35	710M	
hadoop-3.3.6-aarch64.tar.gz.asc	2023-06-25 23:35	833	
hadoop-3.3.6-aarch64.tar.gz.sha512	2023-06-25 23:35	164	
hadoop-3.3.6-rat.txt	2023-06-25 23:35	2.0M	
hadoop-3.3.6-rat.txt.asc	2023-06-25 23:35	833	
hadoop-3.3.6-rat.txt.sha512	2023-06-25 23:35	165	
hadoop-3.3.6-site.tar.gz	2023-06-25 23:35	41M	
hadoop-3.3.6-site.tar.gz.asc	2023-06-25 23:35	833	
hadoop-3.3.6-site.tar.gz.sha512	2023-06-25 23:35	169	
hadoop-3.3.6-src.tar.gz	2023-06-25 23:35	36M	
hadoop-3.3.6-src.tar.gz.asc	2023-06-25 23:35	833	
hadoop-3.3.6-src.tar.gz.sha512	2023-06-25 23:35	168	
hadoop-3.3.6.tar.gz	2023-06-25 23:35	696M	
hadoop-3.3.6.tar.gz.asc	2023-06-25 23:35	833	
hadoop-3.3.6.tar.gz.sha512	2023-06-25 23:35	164	

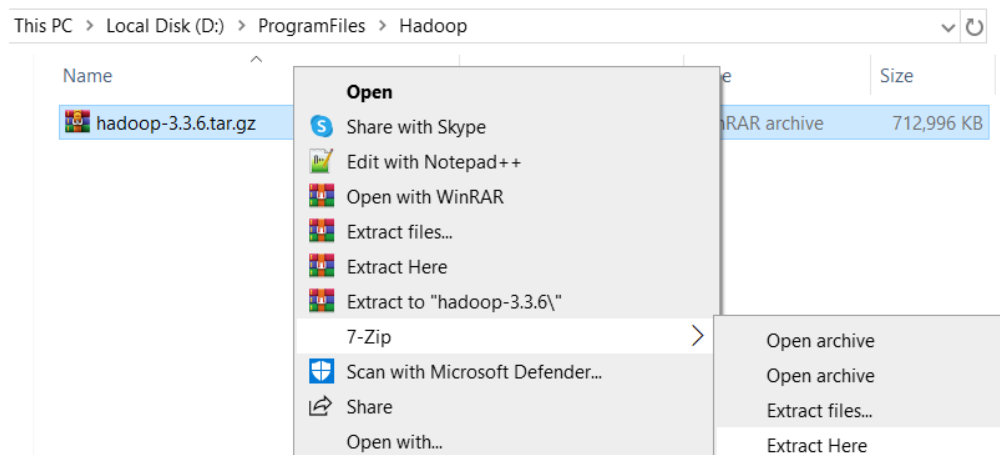
After the binary file is downloaded, unpack it using any file archiver (7zip or WinRAR) utility as below:

- Choose the installation directory in your machine and copy `hadoop-3.3.6.tar.gz` file to that directory. Here, we are choosing Hadoop installation directory as `D:\ProgramFiles\Hadoop`.

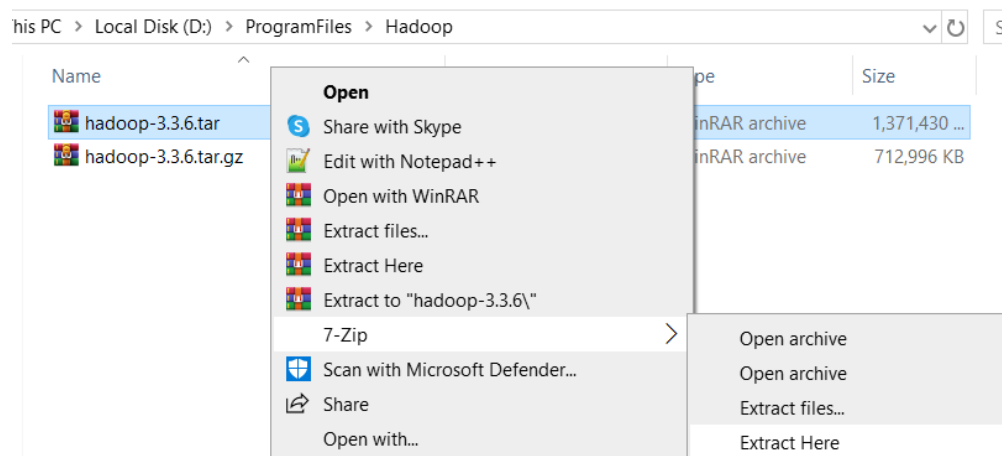
This PC > Local Disk (D:) > ProgramFiles > Hadoop

Name	Date modified	Type	Size
 <code>hadoop-3.3.6.tar.gz</code>	5/24/2024 10:33 PM	WinRAR archive	712,996 KB

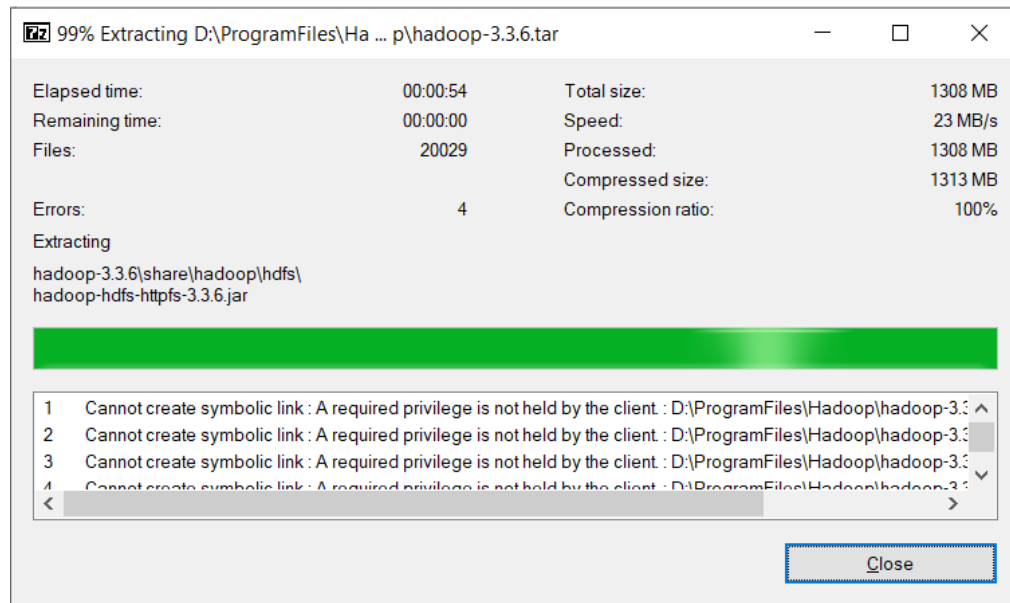
- Right click on the file and choose **7-Zip -> Extract Here** option which extracts a new packed file `hadoop-3.3.6.tar`.



- 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.



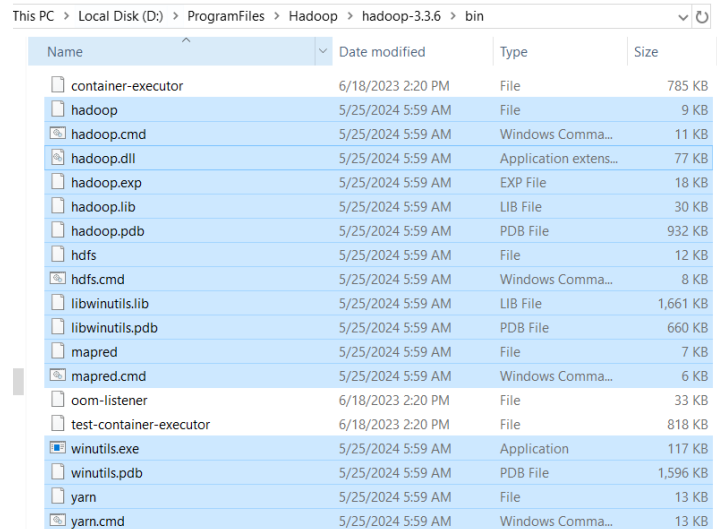
- After the file extraction is completed, you see a folder named `hadoop-3.3.6` which consists of Hadoop binaries and libraries.

PC > Local Disk (D:) > ProgramFiles > Hadoop > hadoop-3.3.6				
Name	Date modified	Type	Size	
bin	6/18/2023 2:38 PM	File folder		
etc	6/18/2023 1:54 PM	File folder		
include	6/18/2023 2:38 PM	File folder		
lib	6/18/2023 2:38 PM	File folder		
libexec	6/18/2023 2:38 PM	File folder		
licenses-binary	6/18/2023 2:38 PM	File folder		
sbin	6/18/2023 1:54 PM	File folder		
share	6/18/2023 3:07 PM	File folder		
LICENSE.txt	6/10/2023 5:11 AM	Text Document	15 KB	
LICENSE-binary	6/14/2023 5:46 AM	File	24 KB	
NOTICE.txt	6/10/2023 5:03 AM	Text Document	2 KB	
NOTICE-binary	6/10/2023 5:11 AM	File	29 KB	
README.txt	6/10/2023 5:03 AM	Text Document	1 KB	

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).



Name	Date modified	Type	Size
container-executor	6/18/2023 2:20 PM	File	785 KB
hadoop	5/25/2024 5:59 AM	File	9 KB
hadoop.cmd	5/25/2024 5:59 AM	Windows Comma...	11 KB
hadoop.dll	5/25/2024 5:59 AM	Application extens...	77 KB
hadoop.exp	5/25/2024 5:59 AM	EXP File	18 KB
hadoop.lib	5/25/2024 5:59 AM	LIB File	30 KB
hadoop.pdb	5/25/2024 5:59 AM	PDB File	932 KB
hdfs	5/25/2024 5:59 AM	File	12 KB
hdfs.cmd	5/25/2024 5:59 AM	Windows Comma...	8 KB
libwinutils.lib	5/25/2024 5:59 AM	LIB File	1,661 KB
libwinutils.pdb	5/25/2024 5:59 AM	PDB File	660 KB
mapred	5/25/2024 5:59 AM	File	7 KB
mapred.cmd	5/25/2024 5:59 AM	Windows Comma...	6 KB
oom-listener	6/18/2023 2:20 PM	File	33 KB
test-container-executor	6/18/2023 2:20 PM	File	818 KB
winutils.exe	5/25/2024 5:59 AM	Application	117 KB
winutils.pdb	5/25/2024 5:59 AM	PDB File	1,596 KB
yarn	5/25/2024 5:59 AM	File	13 KB
yarn.cmd	5/25/2024 5:59 AM	Windows Comma...	13 KB

4. Set up Environment Variables:

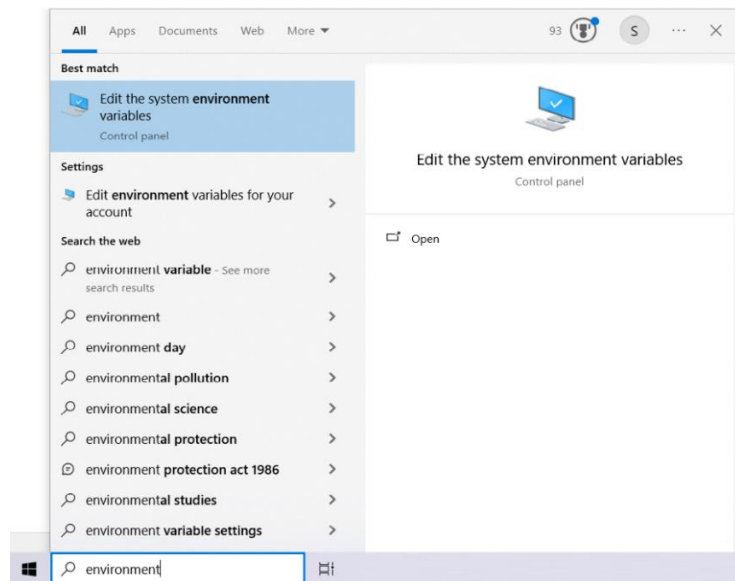
After installing pre-requisites and Hadoop, we should configure the below two environment variables defining Hadoop and Java installation paths.

- **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.
- **HADOOP_HOME:** This is the Hadoop installation directory path in the machine (*in my machine, it is D:\ProgramFiles\Hadoop\hadoop-3.3.6*)

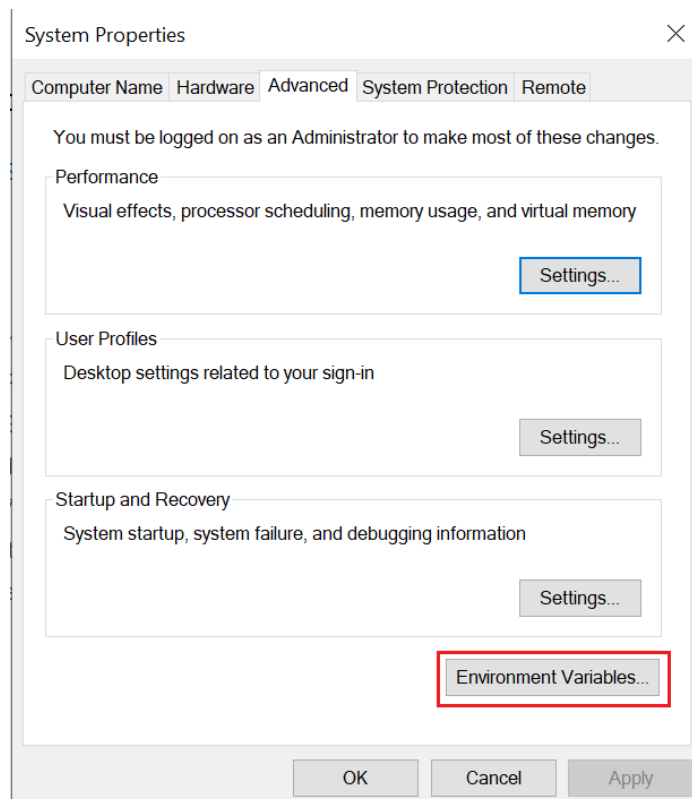
These variables need to be added to either **User environment variables** or **System environment variables** depending on Hadoop configuration needed **for a single user** or **for multiple users**.

In this tutorial, 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.

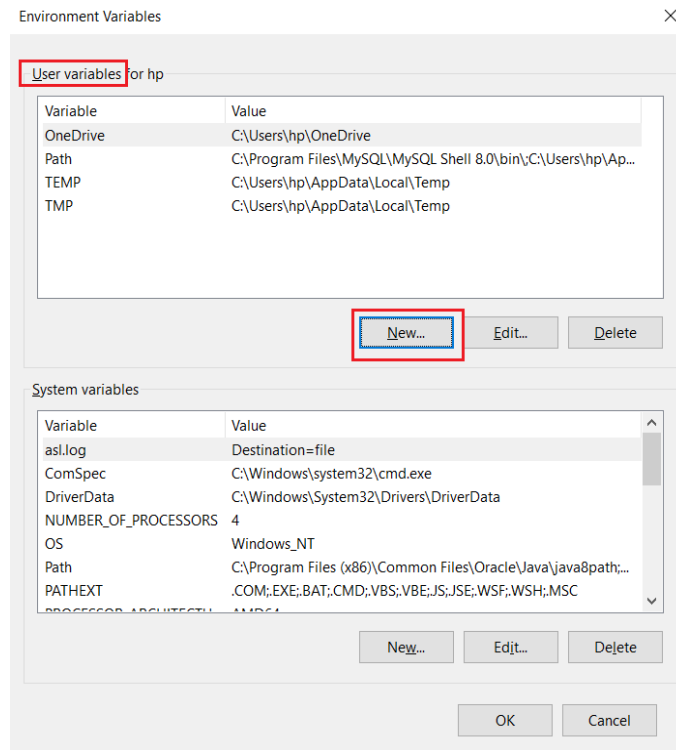
In the Windows search bar, start typing “environment variables” and select the first match which opens up **System Properties** dialog.



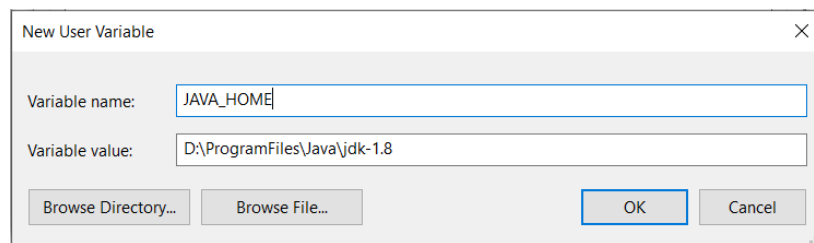
On the **System Properties** window, press **Environment Variables** button.



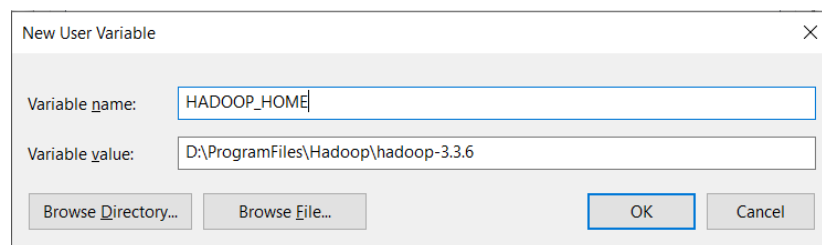
In the **Environment Variables** dialog, click on **New** under **User variables** section.



Add `JAVA_HOME` variable and press OK.

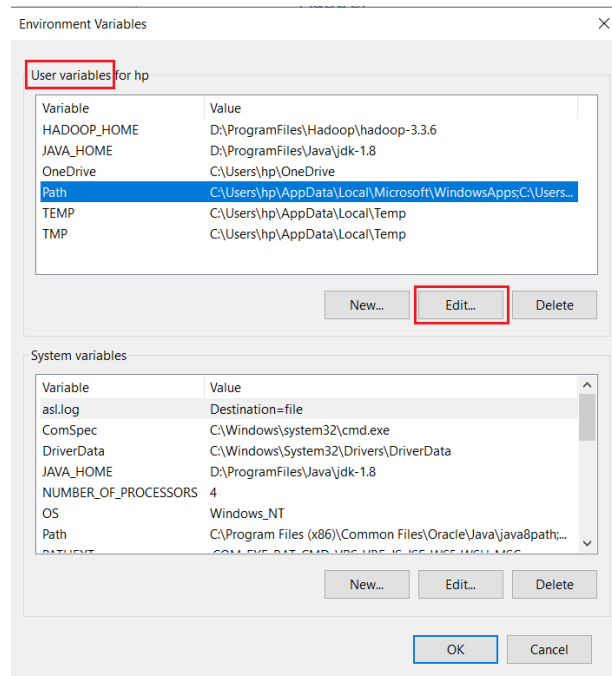


Click on **New** again and add `HADOOP_HOME` variable and press OK.



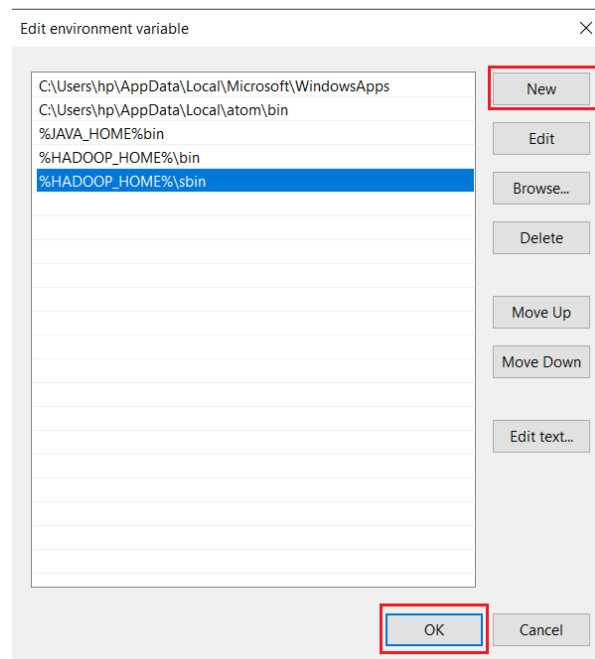
Now, we will update PATH variable to add Java and Hadoop binary paths.

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

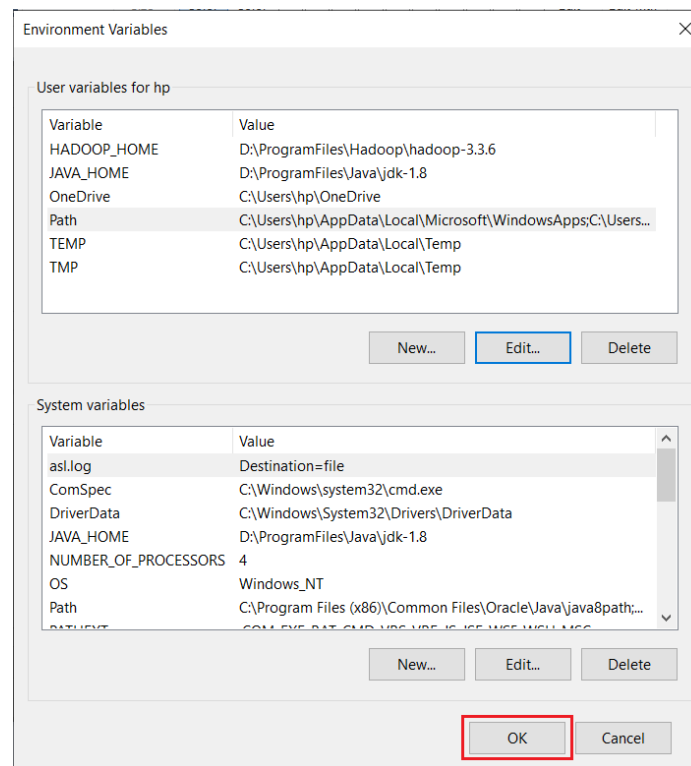


Press **New** and add below values and press OK.

```
%JAVA_HOME%\bin  
%HADOOP_HOME%\bin  
%HADOOP_HOME%\sbin
```



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
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS 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 /D:/ProgramFiles/Hadoop/hadoop-3.3.6/share/hadoop/common/hadoop-common-3.3.6.jar
PS C:\Users\hp>
```

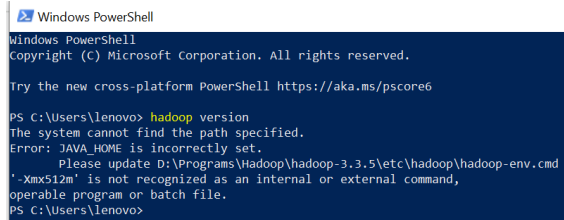
It shows **Hadoop 3.3.6** version which indicates that Hadoop has been installed successfully.

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



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\lenovo> hadoop version
The system cannot find the path specified.
Error: JAVA_HOME is incorrectly set.
Please update D:\Programs\Hadoop\hadoop-3.3.5\etc\hadoop\hadoop-env.cmd
'Xmx512m' is not recognized as an internal or external command,
operable program or batch file.
PS C:\Users\lenovo>
```

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".

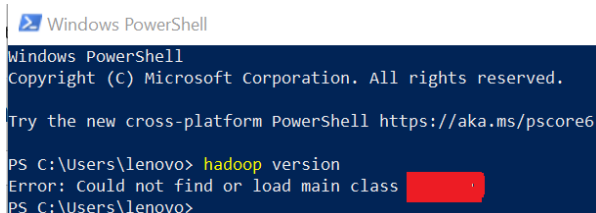
To resolve this issue, use windows 8.3 path in the JAVA_HOME variable. As an example:

- 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



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\lenovo> hadoop version
Error: Could not find or load main class
PS C:\Users\lenovo>
```

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
```

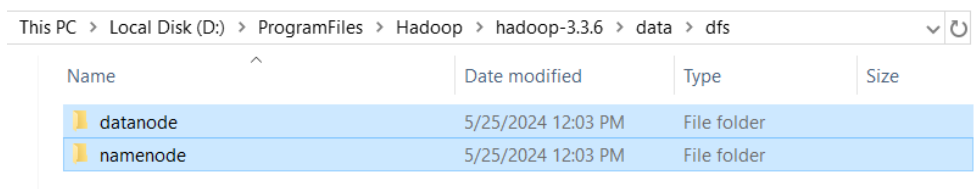
6.1. HDFS Site Configuration:

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 configuration 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 installed 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
```



This PC > Local Disk (D:) > ProgramFiles > Hadoop > hadoop-3.3.6 > data > dfs			
Name	Date modified	Type	Size
datanode	5/25/2024 12:03 PM	File folder	
namenode	5/25/2024 12:03 PM	File folder	

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.

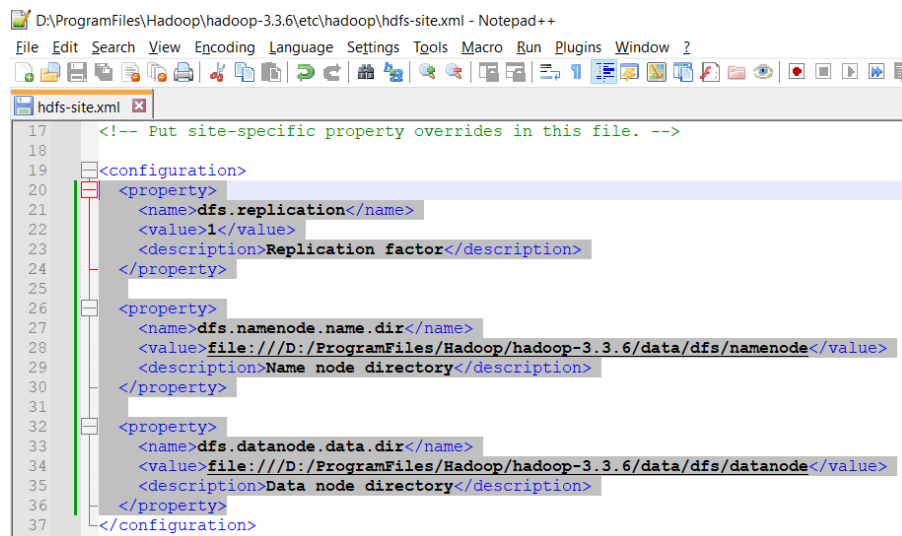
```
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Replication factor</description>
</property>

<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:

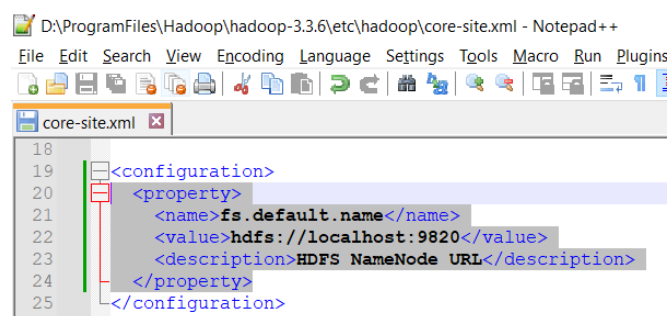
Now, we need to configure the NameNode URL which is `hdfs://localhost:9820`.

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>

```

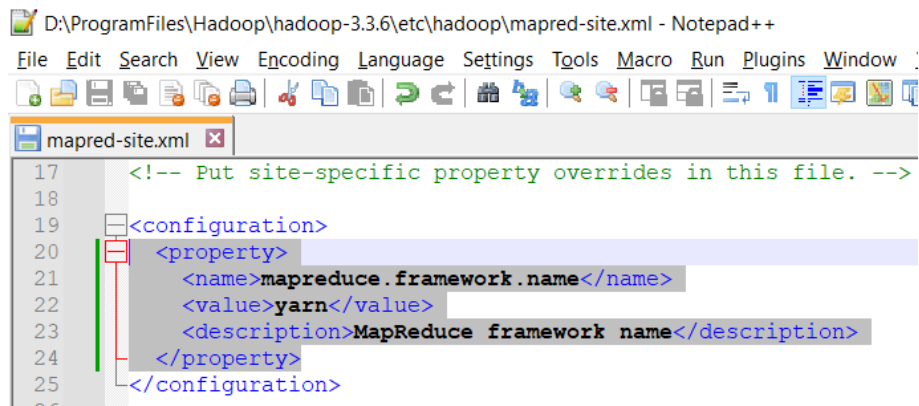


6.3. MapReduce Site Configuration:

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>
```

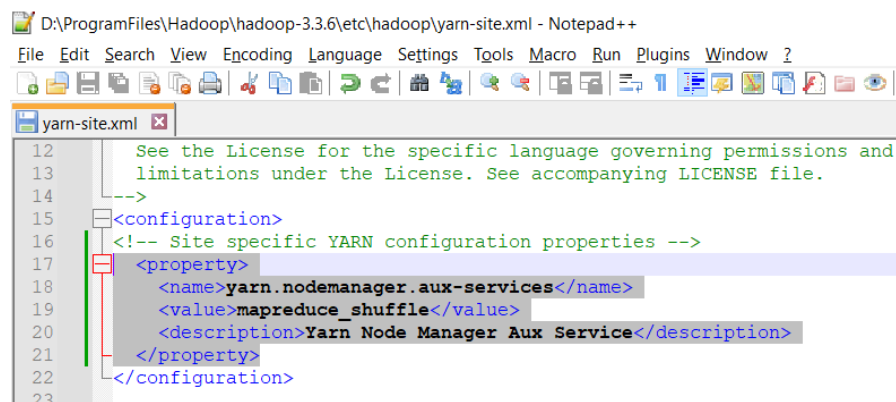


6.4. YARN Site Configuration:

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>
```

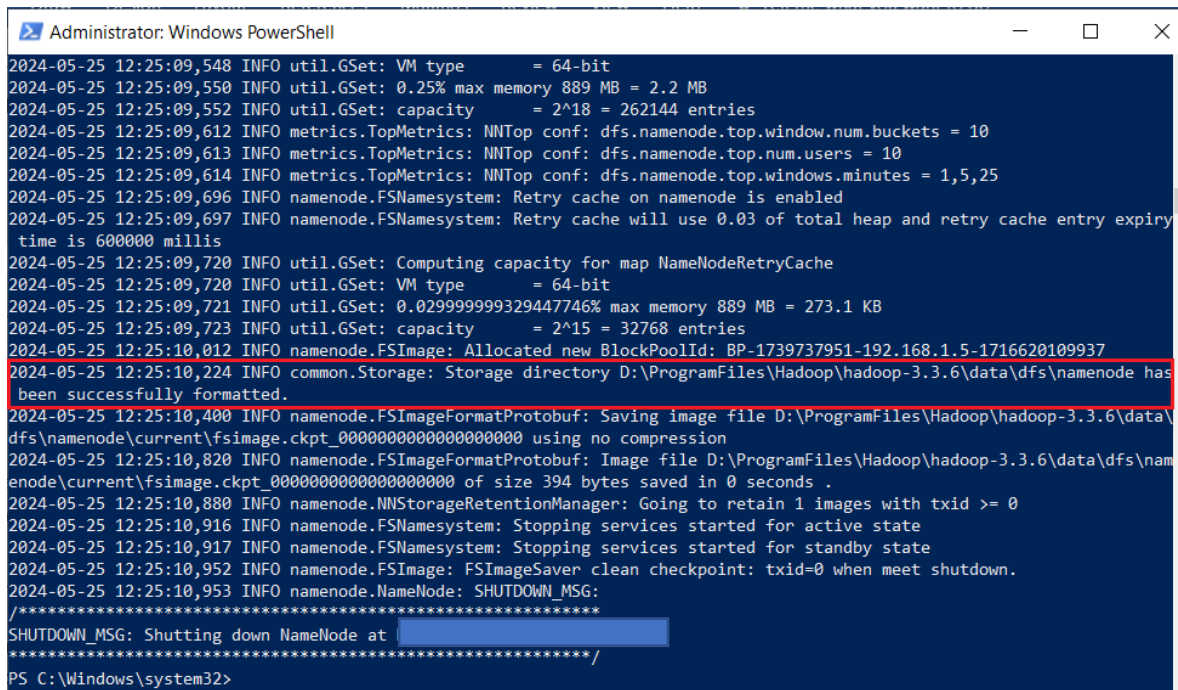


7. Format NameNode:

After completing the Hadoop configuration, it's time to format the NameNode to bring the above configuration changes into effect.

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

```
hdfs namenode -format
```



```
Administrator: Windows PowerShell
2024-05-25 12:25:09,548 INFO util.GSet: VM type = 64-bit
2024-05-25 12:25:09,550 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
2024-05-25 12:25:09,552 INFO util.GSet: capacity = 2^18 = 262144 entries
2024-05-25 12:25:09,612 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
2024-05-25 12:25:09,613 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
2024-05-25 12:25:09,614 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
2024-05-25 12:25:09,696 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
2024-05-25 12:25:09,697 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry
time is 600000 millis
2024-05-25 12:25:09,720 INFO util.GSet: Computing capacity for map NameNodeRetryCache
2024-05-25 12:25:09,720 INFO util.GSet: VM type = 64-bit
2024-05-25 12:25:09,721 INFO util.GSet: 0.0299999999329447746% max memory 889 MB = 273.1 KB
2024-05-25 12:25:09,723 INFO util.GSet: capacity = 2^15 = 32768 entries
2024-05-25 12:25:10,012 INFO namenode.FSImage: Allocated new BlockPoolId: BP-1739737951-192.168.1.5-1716620109937
2024-05-25 12:25:10,224 INFO common.Storage: Storage directory D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\namenode has
been successfully formatted.
2024-05-25 12:25:10,400 INFO namenode.FSImageFormatProtobuf: Saving image file D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\
dfs\namenode\current\fsimage.ckpt_00000000000000000000 using no compression
2024-05-25 12:25:10,820 INFO namenode.FSImageFormatProtobuf: Image file D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\nam
enode\current\fsimage.ckpt_00000000000000000000 of size 394 bytes saved in 0 seconds .
2024-05-25 12:25:10,880 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2024-05-25 12:25:10,916 INFO namenode.FSNamesystem: Stopping services started for active state
2024-05-25 12:25:10,917 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-05-25 12:25:10,952 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2024-05-25 12:25:10,953 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at [redacted]
*****/
PS C:\Windows\system32>
```

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

8. Start Hadoop Services:

Open **Windows PowerShell** or **Command Prompt** as **Administrator** and start services.

8.1. Start Hadoop Nodes:

Run the following command to start the Hadoop nodes.

```
start-dfs.cmd
```

```

Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> start-dfs.cmd
PS C:\Windows\system32>

```

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*”.

```

Apache Hadoop Distribution - hadoop namenode
-
-replicated blocks completed in 10 msec
2024-05-25 12:39:58,603 INFO ipc.Server: IPC Server Responder: starting
2024-05-25 12:39:58,608 INFO ipc.Server: IPC Server listener on 9820: starting
2024-05-25 12:39:58,613 INFO namenode.NameNode: NameNode RPC up at: localhost/127.0.0.1:9820
2024-05-25 12:39:58,618 INFO namenode.FSNamesystem: Starting services required for active state
2024-05-25 12:39:58,619 INFO namenode.FSDirectory: Initializing quota with 12 thread(s)
2024-05-25 12:39:58,632 INFO namenode.FSDirectory: Quota initialization completed in 13 milliseconds
name space=1
storage space=0
storage types=RAM_DISK=0, SSD=0, DISK=0, ARCHIVE=0, PROVIDED=0
2024-05-25 12:39:58,645 INFO blockmanagement.CacheReplicationMonitor: Starting CacheReplicationMonitor with interval 300
00 milliseconds
2024-05-25 12:39:59,566 INFO hdfs.StateChange: BLOCK* registerDatanode: from DatanodeRegistration(127.0.0.1:9866, datanode
deUId=123a143e-536d-495e-a2ae-789991cf74d4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-0
c83702c-4b94-4f10-beca-5173e24efdbc;nsid=195384883;c=1716620109937) storage 123a143e-536d-495e-a2ae-789991cf74d4
2024-05-25 12:39:59,569 INFO net.NetworkTopology: Adding a new node: /default-rack/127.0.0.1:9866
2024-05-25 12:39:59,569 INFO blockmanagement.BlockReportLeaseManager: Registered DN 123a143e-536d-495e-a2ae-789991cf74d4
(127.0.0.1:9866).
2024-05-25 12:39:59,692 INFO blockmanagement.DatanodeDescriptor: Adding new storage ID DS-51797c33-6f1d-4c19-80d1-e4f469
81927d for DN 127.0.0.1:9866
2024-05-25 12:39:59,755 INFO BlockStateChange: BLOCK* processReport 0x997ff9a427e13ff6 with lease ID 0x23159eefe49eaf34:
Processing first storage report for DS-51797c33-6f1d-4c19-80d1-e4f46981927d from datanode DatanodeRegistration(127.0.0.
1:9866, datanodeUId=123a143e-536d-495e-a2ae-789991cf74d4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv
=-57;cid=CID-0c83702c-4b94-4f10-beca-5173e24efdbc;nsid=195384883;c=1716620109937)
2024-05-25 12:39:59,759 INFO BlockStateChange: BLOCK* processReport 0x997ff9a427e13ff6 with lease ID 0x23159eefe49eaf34:
from storage DS-51797c33-6f1d-4c19-80d1-e4f46981927d node DatanodeRegistration(127.0.0.1:9866, datanodeUId=123a143e-53
6d-495e-a2ae-789991cf74d4, infoPort=9864, infoSecurePort=0, ipcPort=9867, storageInfo=lv=-57;cid=CID-0c83702c-4b94-4f10-
beca-5173e24efdbc;nsid=195384883;c=1716620109937), blocks: 0, hasStaleStorage: false, processing time: 4 msec, invalida
tedBlocks: 0

```

```

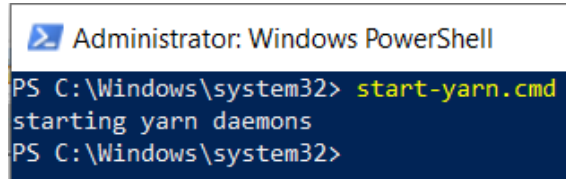
Apache Hadoop Distribution - hadoop datanode
-
2024-05-25 12:39:59,450 INFO impl.FsDatasetImpl: Time to add replicas to map for block pool BP-1739737951-16620109937
on volume D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode: 3ms
2024-05-25 12:39:59,451 INFO impl.FsDatasetImpl: Total time to add all replicas to map for block pool BP-1739737951-192
-1716620109937: 5ms
2024-05-25 12:39:59,454 INFO checker.ThrottledAsyncChecker: Scheduling a check for D:\ProgramFiles\Hadoop\hadoop-3.3.6\da
ta\dfs\datanode
2024-05-25 12:39:59,467 INFO checker.DatasetVolumeChecker: Scheduled health check for volume D:\ProgramFiles\Hadoop\hado
op-3.3.6\data\dfs\datanode
2024-05-25 12:39:59,489 INFO datanode.VolumeScanner: VolumeScanner(D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode
, DS-51797c33-6f1d-4c19-80d1-e4f46981927d): no suitable block pools found to scan. Waiting 1813969544 ms.
2024-05-25 12:39:59,494 WARN datanode.DirectoryScanner: dfs.datanode.directoryscan.throttle.limit.ms.per.sec set to valu
e above 1000 ms/sec. Assuming default value of -1
2024-05-25 12:39:59,494 INFO datanode.DirectoryScanner: Periodic Directory Tree Verification scan starting in 6194606ms
with interval of 21600000ms and throttle limit of -1ms/s
2024-05-25 12:39:59,503 INFO datanode.DataNode: Block pool BP-1739737951-16620109937 (Datanode Uuid 123a14
3e-536d-495e-a2ae-789991cf74d4) service to localhost/127.0.0.1:9820 beginning handshake with NN
2024-05-25 12:39:59,594 INFO datanode.DataNode: Block pool BP-1739737951-16620109937 (Datanode Uuid 123a14
3e-536d-495e-a2ae-789991cf74d4) service to localhost/127.0.0.1:9820 successfully registered with NN
2024-05-25 12:39:59,596 INFO datanode.DataNode: For namenode localhost/127.0.0.1:9820 using BLOCKREPORT_INTERVAL of 2160
0000msecs CACHEREPORT_INTERVAL of 10000msecs Initial delay: 0msecs; heartBeatInterval=3000
2024-05-25 12:39:59,596 INFO datanode.DataNode: Starting IBR Task Handler.
2024-05-25 12:39:59,721 INFO datanode.DataNode: After receiving heartbeat response, updating state of namenode localhost
:9820 to active
2024-05-25 12:39:59,798 INFO datanode.DataNode: Successfully sent block report 0x997ff9a427e13ff6 with lease ID 0x23159e
efe49eaf34 to namenode: localhost/127.0.0.1:9820 containing 1 storage report(s), of which we sent 1. The reports had 0
total blocks and used 1 RPC(s). This took 5 msecs to generate and 70 msecs for RPC and NN processing. Got back one comm
and: FinalizeCommand/5.
2024-05-25 12:39:59,800 INFO datanode.DataNode: Got finalize command for block pool BP-1739737951-16620109
937

```

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"*

```
Apache Hadoop Distribution - yarn resourcemanager
rPB to the server
2024-05-25 12:52:04,985 INFO ipc.Server: IPC Server listener on 8031: starting
2024-05-25 12:52:04,985 INFO ipc.Server: IPC Server Responder: starting
2024-05-25 12:52:04,999 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-05-25 12:52:05,054 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2024-05-25 12:52:05,064 INFO ipc.Server: Listener at 0.0.0.0:8030
2024-05-25 12:52:05,068 INFO ipc.Server: Starting Socket Reader #1 for port 8030
2024-05-25 12:52:05,079 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationMasterProtocolPB to the server
2024-05-25 12:52:05,080 INFO ipc.Server: IPC Server Responder: starting
2024-05-25 12:52:05,080 INFO ipc.Server: IPC Server listener on 8030: starting
2024-05-25 12:52:05,262 INFO ipc.CallQueueManager: Using callQueue: class java.util.concurrent.LinkedBlockingQueue, queueCapacity: 5000, scheduler: class org.apache.hadoop.ipc.DefaultRpcScheduler, ipcBackoff: false.
2024-05-25 12:52:05,263 INFO ipc.Server: Listener at 0.0.0.0:8032
2024-05-25 12:52:05,266 INFO ipc.Server: Starting Socket Reader #1 for port 8032
2024-05-25 12:52:05,273 INFO pb.RpcServerFactoryPBImpl: Adding protocol org.apache.hadoop.yarn.api.ApplicationClientProtocolPB to the server
2024-05-25 12:52:05,277 INFO ipc.Server: IPC Server listener on 8032: starting
2024-05-25 12:52:05,279 INFO ipc.Server: IPC Server Responder: starting
2024-05-25 12:52:05,381 INFO resourcemanager.ResourceTrackerService: NodeManager from node DESKTOP-KGH2E2G(cmPort: 14104, httpPort: 8042) registered with capability: <memory:8192, vCores:8>, assigned nodeId DESKTOP-KGH2E2G:14104
2024-05-25 12:52:05,386 INFO rmnode.RMNodeImpl: DESKTOP-KGH2E2G:14104 Node Transitioned from NEW to RUNNING
2024-05-25 12:52:05,445 INFO capacity.CapacityScheduler: Added node DESKTOP-KGH2E2G:14104 clusterResource: <memory:8192, vCores:8>
2024-05-25 12:52:06,621 INFO webproxy.ProxyCA: Created Certificate for OU=YARN-e96a8c82-1c05-44a1-89eb-5efa7db36b62
2024-05-25 12:52:06,743 INFO recovery.RMStateStore: Storing CA Certificate and Private Key
2024-05-25 12:52:06,744 INFO resourcemanager.ResourceManager: Transitioned to active state
```

```
Apache Hadoop Distribution - yarn nodemanager
May 25, 2024 12:52:01 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory register
INFO: Registering org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver as a provider class
May 25, 2024 12:52:01 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.19.4 05/24/2017 03:20 PM'
May 25, 2024 12:52:01 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.JAXBContextResolver to GuiceManagedComponentProvider with the scope "Singleton"
May 25, 2024 12:52:02 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.webapp.GenericExceptionHandler to GuiceManagedComponentProvider with the scope "Singleton"
May 25, 2024 12:52:03 PM com.sun.jersey.guice.spi.container.GuiceComponentProviderFactory getComponentProvider
INFO: Binding org.apache.hadoop.yarn.server.nodemanager.webapp.NMWebServices to GuiceManagedComponentProvider with the scope "Singleton"
2024-05-25 12:52:03,580 INFO handler.ContextHandler: Started o.e.j.w.WebAppContext@7a583586{node/,file:///C:/Users/hp/AppData/Local/Temp/jetty-0_0_0-8042-hadoop-yarn-common-3_3_6_jar-_-any-8969605726863143894/webapp/,AVAILABLE}{jar:file:/D:/ProgramFiles/Hadoop/hadoop-3.3.6/share/hadoop/yarn/hadoop-yarn-common-3.3.6.jar!/webapps/node}
2024-05-25 12:52:03,599 INFO server.AbstractConnector: Started ServerConnector@23c388c2[HTTP/1.1, (http/1.1)]{0.0.0.0:8042}
2024-05-25 12:52:03,601 INFO server.Server: Started @12826ms
2024-05-25 12:52:03,601 INFO webapp.WebApps: Web app node started at 8042
2024-05-25 12:52:03,604 INFO nodemanager.NodeStatusUpdaterImpl: Node ID assigned is : DESKTOP-KGH2E2G:14104
2024-05-25 12:52:03,612 INFO util.JvmPauseMonitor: Starting JVM pause monitor
2024-05-25 12:52:03,628 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8031
2024-05-25 12:52:05,423 INFO security.NMContainerTokenSecretManager: Rolling master-key for container-tokens, got key with id -341551026
2024-05-25 12:52:05,427 INFO security.NMTokenSecretManagerInNM: Rolling master-key for container-tokens, got key with id -944524098
2024-05-25 12:52:05,428 INFO nodemanager.NodeStatusUpdaterImpl: Registered with ResourceManager as DESKTOP-KGH2E2G:14104 with total resource of <memory:8192, vCores:8>
```

Note:

You can start all the above 4 services using a single command as below

```
start-all.cmd
```

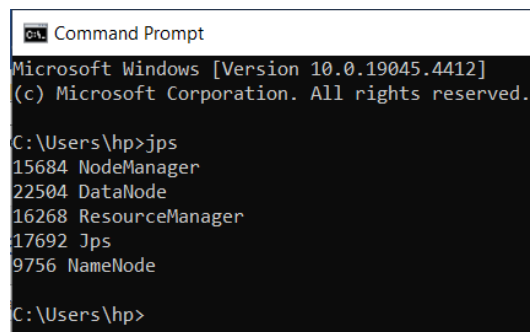
8.3. Verify Services:

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
```

A screenshot of a Windows Command Prompt window. The title bar reads 'C:\> Command Prompt'. The window content shows the Microsoft Windows version (10.0.19045.4412) and copyright information. The user is at the prompt 'C:\Users\hp>' and has entered the command 'jps'. The output of the command is displayed as follows:

```
C:\Users\hp>jps
15684 NodeManager
22504 DataNode
16268 ResourceManager
17692 Jps
9756 NameNode
C:\Users\hp>
```

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.

```
Command Prompt
C:\Users\hp>hdfs fsck /
Connecting to namenode via http://localhost:9870/fsck?ugi=hp&path=%2F
FSCK started by hp (auth:SIMPLE) from /127.0.0.1 for path / at Sat May 25 13:03:10 IST 2024

Status: HEALTHY
  Number of data-nodes: 1
  Number of racks:      1
  Total dirs:           1
  Total symlinks:       0

Replicated Blocks:
  Total size: 0 B
  Total files: 0
  Total blocks (validated): 0
  Minimally replicated blocks: 0
  Over-replicated blocks: 0
  Under-replicated blocks: 0
  Mis-replicated blocks: 0
  Default replication factor: 1
  Average block replication: 0.0
  Missing blocks: 0
  Corrupt blocks: 0
  Missing replicas: 0
  Blocks queued for replication: 0

Erasure Coded Block Groups:
  Total size: 0 B
  Total files: 0
  Total block groups (validated): 0
  Minimally erasure-coded block groups: 0
  Over-erasure-coded block groups: 0
  Under-erasure-coded block groups: 0
  Unsatisfactory placement block groups: 0
  Average block group size: 0.0
  Missing block groups: 0
  Corrupt block groups: 0
  Missing internal blocks: 0
  Blocks queued for replication: 0
FSCK ended at Sat May 25 13:03:10 IST 2024 in 27 milliseconds

The filesystem under path '/' is HEALTHY

C:\Users\hp>
```

9.2. List Contents:

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

```
hadoop fs -ls /
```

or

```
hdfs dfs -ls /
```

9.3. Create Directory:

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

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

or

```
hdfs dfs -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
```

or

```
hdfs dfs -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>
```

or

```
hdfs dfs -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>
```

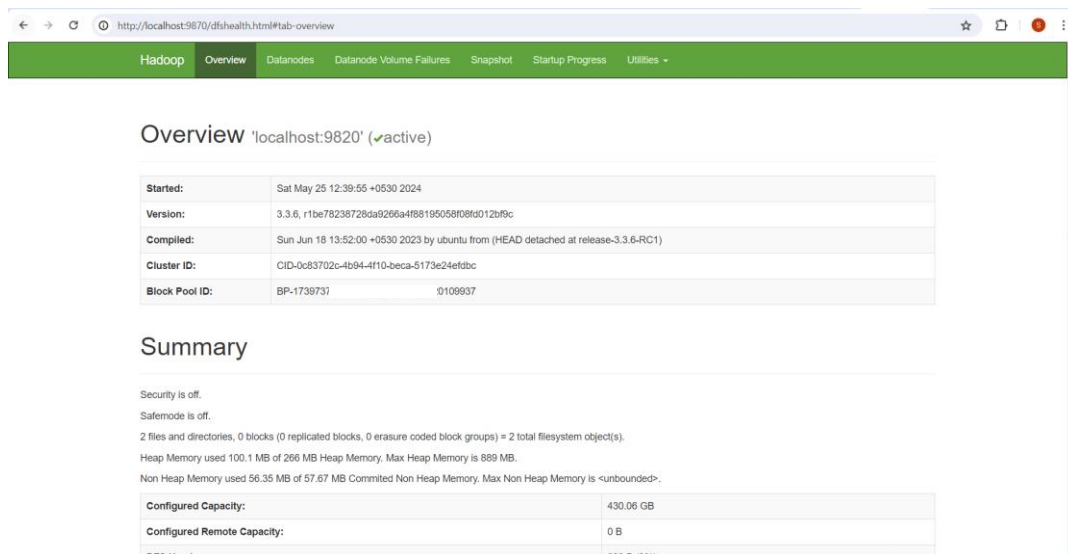
Similarly, we can execute any other HDFS commands on our cluster.

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>



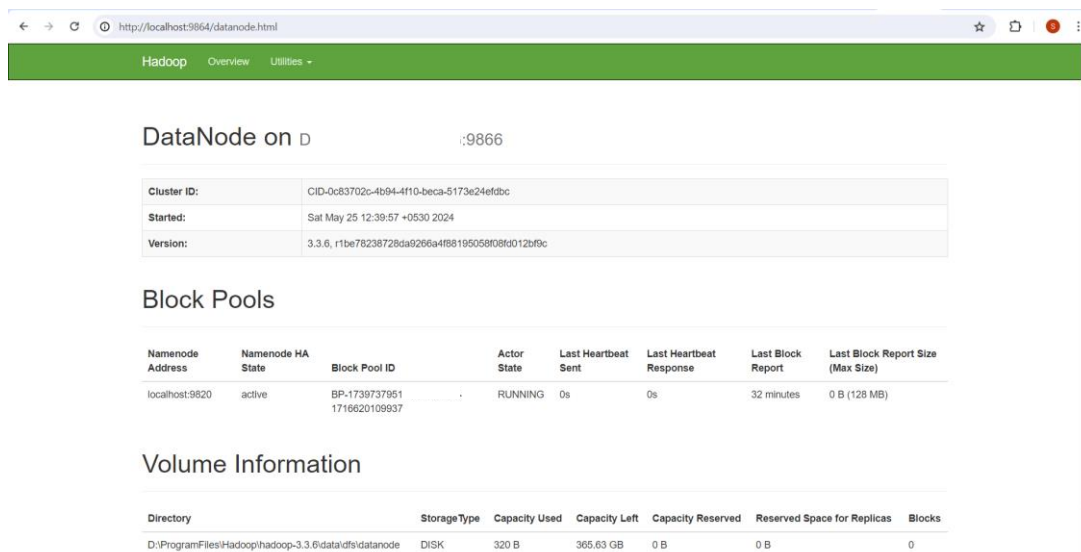
Overview 'localhost:9820' (active)	
Started:	Sat May 25 12:39:55 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012b93c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-0c83702c-4b94-4f10-beca-5173e24efdbc
Block Pool ID:	BP-17397371-1716620109937

Summary	
Security is off.	
Safemode is off.	
2 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 2 total filesystem object(s).	
Heap Memory used 100.1 MB of 266 MB Heap Memory. Max Heap Memory is 889 MB.	
Non Heap Memory used 56.35 MB of 57.67 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.	

Configured Capacity:	
	430.06 GB

Configured Remote Capacity:	
	0 B

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



DataNode on D:9866	
Cluster ID:	CID-0c83702c-4b94-4f10-beca-5173e24efdbc
Started:	Sat May 25 12:39:57 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012b93c

Block Pools							
NameNode Address	NameNode HA State	Block Pool ID	Actor State	Last Heartbeat Sent	Last Heartbeat Response	Last Block Report	Last Block Report Size (Max Size)
localhost:9820	active	BP-1739737951-1716620109937	RUNNING	0s	0s	32 minutes	0 B (128 MB)

Volume Information						
Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
D:\ProgramFiles\Hadoop\hadoop-3.3.6\data\dfs\datanode	DISK	320 B	365.63 GB	0 B	0 B	0

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

The screenshot shows the Hadoop YARN UI at <http://localhost:8088/cluster>. The interface includes a sidebar with navigation links: Cluster, About, Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler, and Tools. The main content area displays various metrics:

- Cluster Metrics:** Apps Submitted (0), Apps Pending (0), Apps Running (0), Apps Completed (0), Containers Running (0), Used Resources (<memory:0 B, vCores:0>), and Maximum Allocated Resources (<memory:8192, vCores:4>).
- Cluster Nodes Metrics:** Active Nodes (1), Decommissioning Nodes (0), Decommissioned Nodes (0), and Lost Nodes (0).
- Scheduler Metrics:** Scheduler Type (Capacity Scheduler), Scheduling Resource Type (memory-mb (unit=M), vcores), Minimum Allocation (<memory:1024, vCores:1>), and Maximum Allocation (<memory:8192, vCores:4>).

Below the metrics, there is a table with columns: ID, User, Name, Application Type, Application Tags, Queue, Application Priority, StartTime, LaunchTime, FinishTime, State, FinalStatus, Running Containers, and Allocated CPU VCore. The table currently shows "No data available in table" and "Showing 0 to 0 of 0 entries".

11. MapReduce Examples:

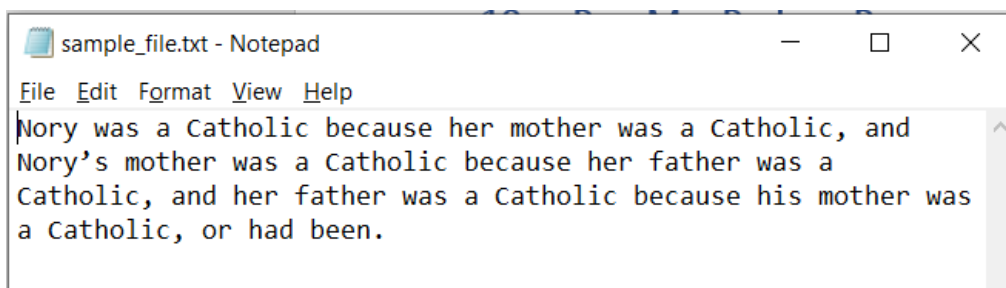
Hadoop MapReduce is a software framework built for writing applications to process huge amounts of data (*multi-terabyte data-sets*) in-parallel on large clusters (*thousands of nodes*) of commodity hardware in a reliable, fault-tolerant manner.

Hadoop 3.3.6 version provides in-built MapReduce example programs such as wordcount, wordmean, aggregatwordcount, 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_1716621724430_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
NoryÇÖs       1
a             6
and           2
because       3
been.         1
father        2
had           1
her           3
his           1
mother        3
or            1
was           6

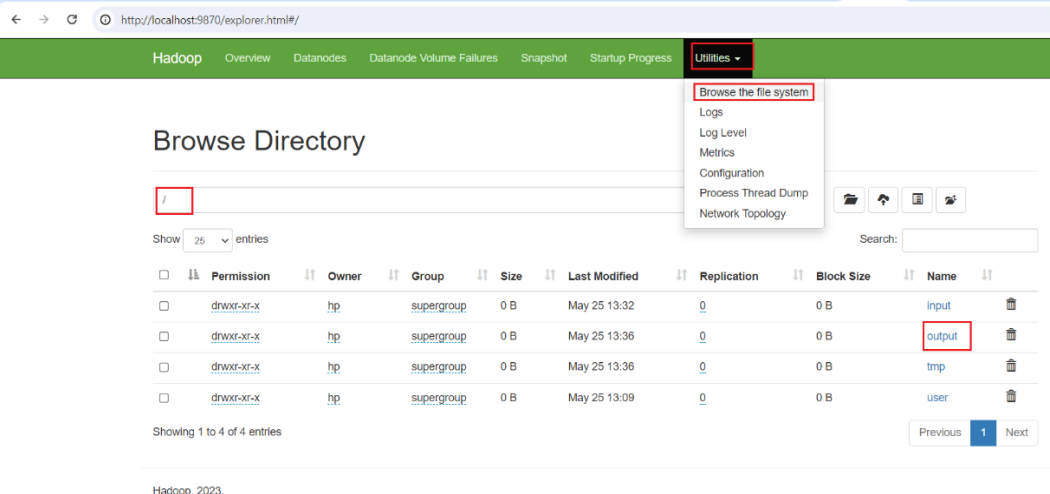
D:\Big Data\Datasets>
```

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. View in 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.



The screenshot shows the Hadoop NameNode UI at <http://localhost:9870/explorer.html#/>. The 'Utilities' tab is selected, and the 'Browse the file system' option is highlighted in the dropdown menu. The main content area shows a table of files and directories under the root directory. The 'output' directory is highlighted in red.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	hp	supergroup	0 B	May 25 13:32	0	0 B	input
drwxr-xr-x	hp	supergroup	0 B	May 25 13:36	0	0 B	output
drwxr-xr-x	hp	supergroup	0 B	May 25 13:36	0	0 B	tmp
drwxr-xr-x	hp	supergroup	0 B	May 25 13:09	0	0 B	user

Then, click on `wordcount` folder.

The screenshot shows the Hadoop Explorer interface at `http://localhost:9870/explorer.html#/output`. The breadcrumb path is `/output`. The table lists the contents of the `/output` directory:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
<code>drwxr-xr-x</code>	<code>hp</code>	<code>supergroup</code>	0 B	May 25 13:36	0	0 B	<code>wordcount</code>

Showing 1 to 1 of 1 entries. The `wordcount` folder is highlighted with a red box.

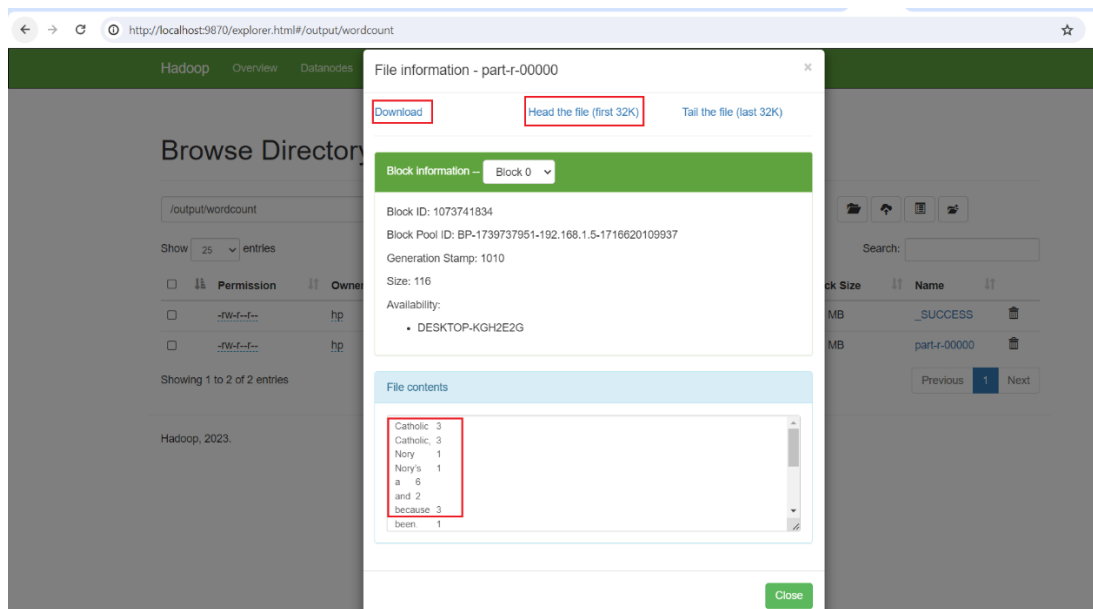
Then click on `part-r-00000` file.

The screenshot shows the Hadoop Explorer interface at `http://localhost:9870/explorer.html#/output/wordcount`. The breadcrumb path is `/output/wordcount`. The table lists the contents of the `/output/wordcount` directory:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
<code>-rw-r--r--</code>	<code>hp</code>	<code>supergroup</code>	0 B	May 25 13:36	1	128 MB	<code>_SUCCESS</code>
<code>-rw-r--r--</code>	<code>hp</code>	<code>supergroup</code>	116 B	May 25 13:36	1	128 MB	<code>part-r-00000</code>

Showing 1 to 2 of 2 entries. The `part-r-00000` file is highlighted with a red box.

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. View 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.

hadoop

All Applications

Cluster

- About
- Nodes
- Node Labels
- Applications
- NEW
- NEW SAVING
- SUBMITTED
- ACCEPTED
- RUNNING
- FINISHED
- FAILED
- KILLED
- Scheduler
- Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources
1	0	0	1	0	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCores:1>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalState
application_1716621724430_0001	hp	word count	MAPREDUCE		default	0	Sat May 25 13:36:22 +0550 2024	Sat May 25 13:36:24 +0550 2024	Sat May 25 13:36:52 +0550 2024	FINISHED	SUCCEEDED

Showing 1 to 1 of 1 entries

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

The screenshot shows the Hadoop Application Overview page for application_1716621724430_0001. The page is divided into several sections:

- Cluster:** A sidebar menu with options like About, Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, Scheduler, and Tools.
- Application Overview:** A table showing application details such as User (hdp), Name (word count), Application Type (MAPREDUCE), Application Tags, Application Priority (0), YarnApplicationState (FINISHED), Queue (default), FinalStatus Reported by AM (SUCCEEDED), Started (Sat May 25 13:36:22 +0530 2024), Launched (Sat May 25 13:36:24 +0530 2024), Finished (Sat May 25 13:36:52 +0530 2024), Elapsed (30sec), Tracking URL (History), Log Aggregation Status (DISABLED), Application Timeout (Remaining Time) (Unlimited), Diagnostics (false), Unmanaged Application (false), Application Node Label expression (<Not set>), and AM container Node Label expression (<DEFAULT_PARTITION>).
- Application Metrics:** A table showing resource metrics such as Total Resource Preempted (<memory:0, vCores:0>), Total Number of Non-AM Containers Preempted (0), Total Number of AM Containers Preempted (0), Resource Preempted from Current Attempt (<memory:0, vCores:0>), Number of Non-AM Containers Preempted from Current Attempt (0), Aggregate Resource Allocation (85153 MB-seconds, 46 vcore-seconds), and Aggregate Preempted Resource Allocation (0 MB-seconds, 0 vcore-seconds).
- Attempts Table:** A table with columns: Attempt ID, Started, Node, Logs, Nodes blacklisted by the app, and Nodes blacklisted by the system. It shows one attempt: appattempt_1716621724430_0001_000001, started on Sat May 25 13:36:22 +0550 2024, on node http://DESKTOP-KGH2EZG.8042, with 0 logs, 0 nodes blacklisted by the app, and 0 nodes blacklisted by the system.

12. Stop Hadoop Services:

Open **Windows PowerShell** or **Command Prompt** as **Administrator** and stop services.

12.1. Stop Hadoop Nodes:

Use the following command to stop the Hadoop nodes.

```
stop-dfs.cmd
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

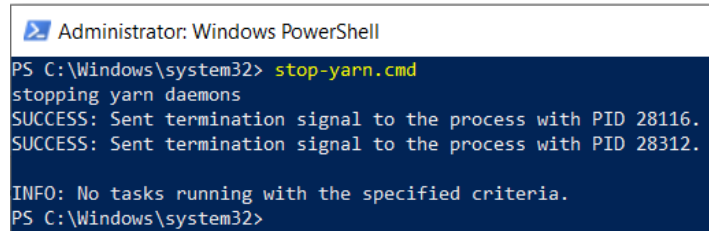
PS C:\Windows\system32> stop-dfs.cmd
SUCCESS: Sent termination signal to the process with PID 24124.
SUCCESS: Sent termination signal to the process with PID 25232.
PS C:\Windows\system32>
```

After executing the above command, it automatically closes two command prompt windows that were opened earlier for the **namenode** and **datanode** daemons.

12.2. Stop Hadoop YARN:

Use the following command to stop Hadoop YARN services

```
stop-yarn.cmd
```



```
Administrator: Windows PowerShell
PS C:\Windows\system32> stop-yarn.cmd
stopping yarn daemons
SUCCESS: Sent termination signal to the process with PID 28116.
SUCCESS: Sent termination signal to the process with PID 28312.

INFO: No tasks running with the specified criteria.
PS C:\Windows\system32>
```

After executing the above command, it automatically closes two command prompt windows that were opened earlier for the **resourcemanager** and **nodemanager** daemons.

Note:

You can stop all the above 4 services using a single command as below

```
stop-all.cmd
```

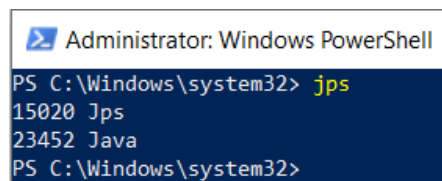
12.3. Verify Services:

Run the following command to make sure all services are stopped successfully:

```
jps
```

It shouldn't display any of the below services:

```
NodeManager
DataNode
ResourceManager
NameNode
```



```
Administrator: Windows PowerShell
PS C:\Windows\system32> jps
15020 Jps
23452 Java
PS C:\Windows\system32>
```

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