

## Introduction

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. For storing and analyzing big data, Hadoop is used. Data is stored in Hadoop on inexpensive commodity servers operating as clusters. It is a distributed file system that enables processing and fault tolerance simultaneously. For faster storage and recovery of data from its nodes, the Hadoop MapReduce programming model is used.

## How to install Hadoop

Step 1: <https://hadoop.apache.org/releases.html>

Step 2: choose 3.3.0 and click the binary

Step 3: After clicking The binary click the

<https://downloads.apache.org/hadoop/common/hadoop-3.3.0/hadoop-3.3.0.tar.gz>

Step 4: Before you install the hadoop you must install the JDK.

Link: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

Step 5: Install the JDK First Then Move The Jdk Folder to local C.

Step 6: Now go to Environment variable for the setup. Click the Environment Variable then click the New button then Input the Variable Name:(it depends on you) Variable Value:(the location of file/ Path of the folder).

Step 7: In System Variable Click the Path Column then paste the (Path of the file) also.

Step 8: Do this in CMD: cd Program Files\Java\jdk-13.0.2\bin

then type javac.

Step 9: After You installed the JDK u must install now the hadoop file. First you need to extract the file then after you extracted the Zip file. now move the file to the local C.

Step 10: After Extracted the Zip file of hadoop. Now go to ETC folder then find the.

```
-- core-site-- xml mode
-- hadoop-env-- cmd mode
-- mapred-site -- xml mode
-- yarn site -- xml mode
-- hdfs site - xml mode
```

Step 11: Create folder inside the hadoop folder (data).

Step 12: Open the folder that you created then create folder again for (datanode and namenode).

## Importance of Hadoop

Hadoop is an open source platform that provides excellent data management provision. It is a framework that supports the processing of large data sets in a distributed computing environment. It is designed to expand from single servers to thousands of machines, each providing computation and storage. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating uninterrupted in case of a node failure, which minimizes the risk of catastrophic system failure, even if a significant number of nodes become out of action. Hadoop is very valuable for large scale businesses basing on its proven benefits for enterprises given below:

Advantages for Enterprises:

- Hadoop provides a cost effective storage solution for business.
- It facilitates businesses to easily access new data sources and tap into different types of data to produce value from that data.
- It is a highly scalable storage platform.
- Unique storage method of Hadoop is based on a distributed file system that basically 'maps' data wherever it is located on a cluster. The tools for data processing are often on the same servers where the data is located, resulting in much faster data processing.
- Hadoop is now widely used across industries, including finance, media and entertainment, government, healthcare, information services, retail, and other industries

## Stored Data to Data Node

If you successfully connected to Hadoop then do this command.

Step 1: C:\Users\artam>hadoop fs -mkdir /sample\_dir (ex jem\_dir)

Step 2: C:\Users\artam>hadoop fs -put [File\_location\_that\_you\_want\_to\_stored] /sample\_dir

Step 3: C:\Users\artam>hadoop fs -cat /sample\_dir/[File\_location\_that\_you\_want\_to\_stored]

Explanation:

Step 1: You are making a file directory in DATANODE.

Step 2: Inserting File to the sample\_dir where sample\_dir is a folder in DATANODE.

Step 3: Display Data from DATANODE

### Other Command:

- C:\Users\artam> hadoop dfsadmin -safemode leave
- C:\Users\artam> hadoop dfsadmin -safemode enter
- C:\Users\artam> hadoop fs -rm -r /input\_dir/sample.txt
- C:\Users\artam> hadoop fs -rm -r /input\_dir

### Explanation:

- You are leaving in a Safe Mode.
- You are Entering in a Safe mode
- Deleting file/data inside the DATANODE where you pointing to inside the input\_dir.
- Deleting input\_dir where input\_dir is a folder of DATANODE.

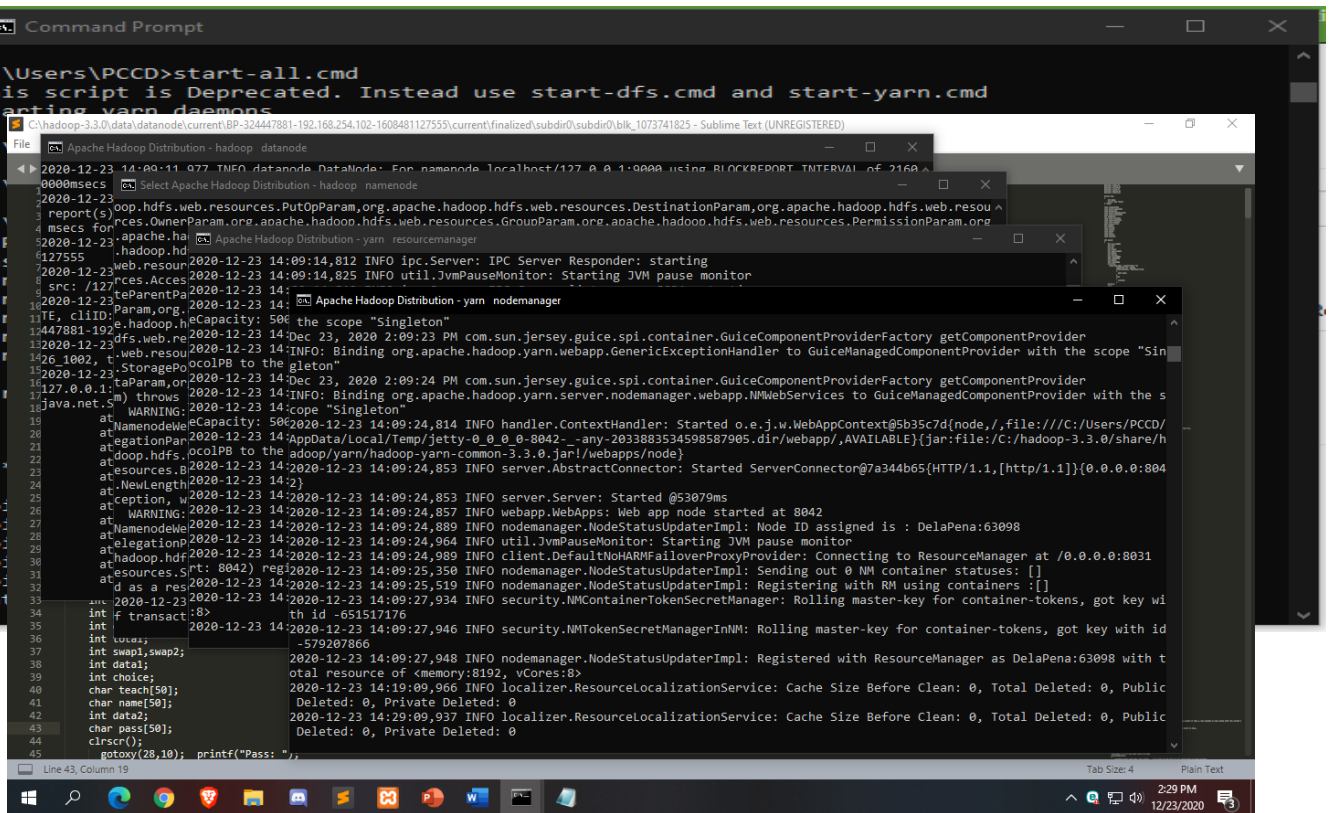
The screenshot shows the Hadoop web interface for browsing the directory /jemwel\_dir. The interface includes a search bar, a 'Go!' button, and a table of files. The table has columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. Two files are listed: 'l.c' and 'jemwel.txt'.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	PCCD	supergroup	14.97 KB	Dec 21 16:29	1	128 MB	l.c
-rw-r--r--	PCCD	supergroup	12 B	Dec 23 14:12	1	128 MB	jemwel.txt

Showing 1 to 2 of 2 entries


Previous 1 Next

Hadoop, 2020.



```
C:\hadoop-3.3.0\data\datanode\current\BP-324447881-192.168.254.102-1608481127555\current\finalized\subdir0\subdir0\blk_1073741825 - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

blk_1073741825
1 #include <stdio.h>
2 #include <string.h>
3 #include <conio.h>
4 #include <malloc.h>
5 #include <stdlib.h>
6
7 struct node
8 {
9     int data;
10    struct node *next;
11 } *head;
12
13 void create(int);
14 void insert_beg(int);
15 void display();
16 void insert_mid(int,int);
17 void insert_end(int);
18 int length();
19 void erase_first();
20 void erase_mid(int);
21 void erase_last();
22 void erase_all();
23 int swap();
24 void sort();
25 void line();
26 void box();
27 void bo();
28 void box1();
29 void line1();
30
31 int main()
32 {
33     int a,b,c,data;
34     int pos;
35     int qw;
36     int total;
37     int swap1,swap2;
38     int data1;
39     int choice;
40     char teach[50];
41     char name[50];
42     int data2;
43     char pass[50];
44     clrscr();
45     gotoxy(28,10); printf("Pass: ");
```



Cluster

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

Tools

## All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved
0	0	0	0	0	0 B	8 GB	0 B

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[memory-mb (unit-Mi), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU Vcores	Allocated Memory MB	Reserved CPU Vcores
No data available in table															

Showing 0 to 0 of 0 entries