EXPERIMENT:

Install, Configure and Run Hadoop and HDFS

<u>PROGRAM:</u>

<u>AIM</u>: To Installing and Running Applications On Hadoop and HDFS.

HADOOP INSTALATION IN WINDOWS

1. Prerequisites

Hardware Requirement

* RAM — Min. 8GB, if you have SSD in your system then 4GB RAM would also work.
* CPU — Min. Quad core, with at least 1.80GHz

2. JRE 1.8 — Offline installer for JRE

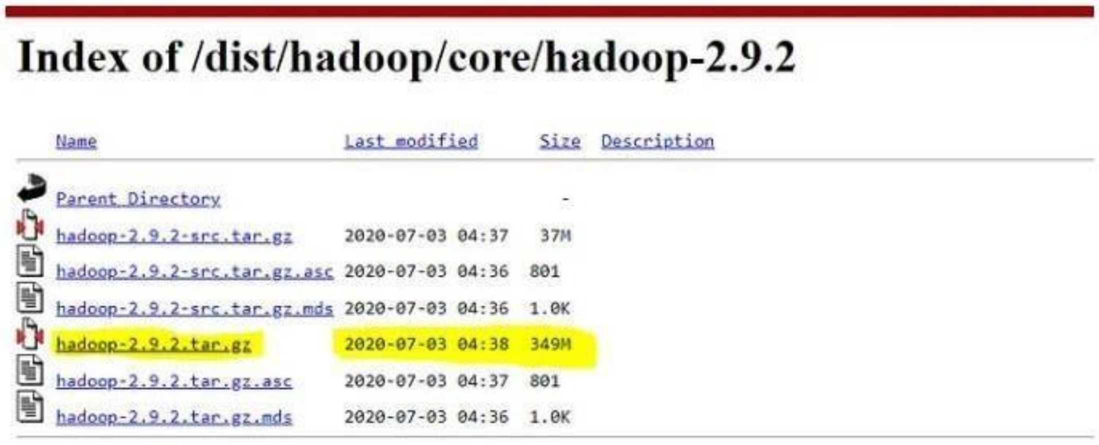
3. Java Development Kit — 1.8

4. A Software for Un-Zipping like 7Zip or Win Rar

* I will be using a 64-bit windows for the process, please check and download the version supported by your system x86 or x64 for all the software.

5. Download Hadoop zip

* I am using Hadoop-2.9.2, you can use any other STABLE version for hadoop.



```
Index of /dist/hadoop/core/hadoop-2.9.2

Name                         Last modified      Size  Description

Parent Directory                                  -
hadoop-2.9.2-src.tar.gz      2020-07-03 04:37   37M
hadoop-2.9.2-src.tar.gz.asc  2020-07-03 04:36   801
hadoop-2.9.2-src.tar.gz.mds  2020-07-03 04:36   1.0K
hadoop-2.9.2.tar.gz          2020-07-03 04:38   349M
hadoop-2.9.2.tar.gz.asc      2020-07-03 04:37   801
hadoop-2.9.2.tar.gz.mds      2020-07-03 04:36   1.0K
```

Fig. 1:- Download Hadoop 2.9.2

Once we have Downloaded all the above software, we can proceed with next steps in installing the Hadoop.

2. Unzip and Install Hadoop

After Downloading the Hadoop, we need to Unzip the hadoop-2.9.2.tar.gz file.

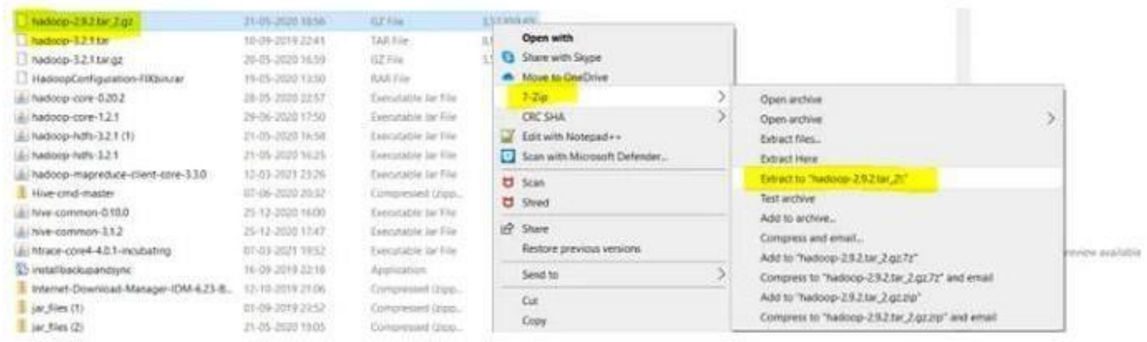
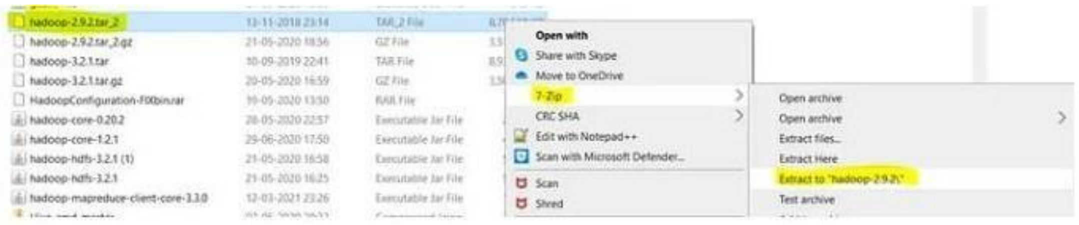


Fig. 2:- Extracting Hadoop Step-1

Once extracted, we would get a new file hadoop-2.9.2.tar.

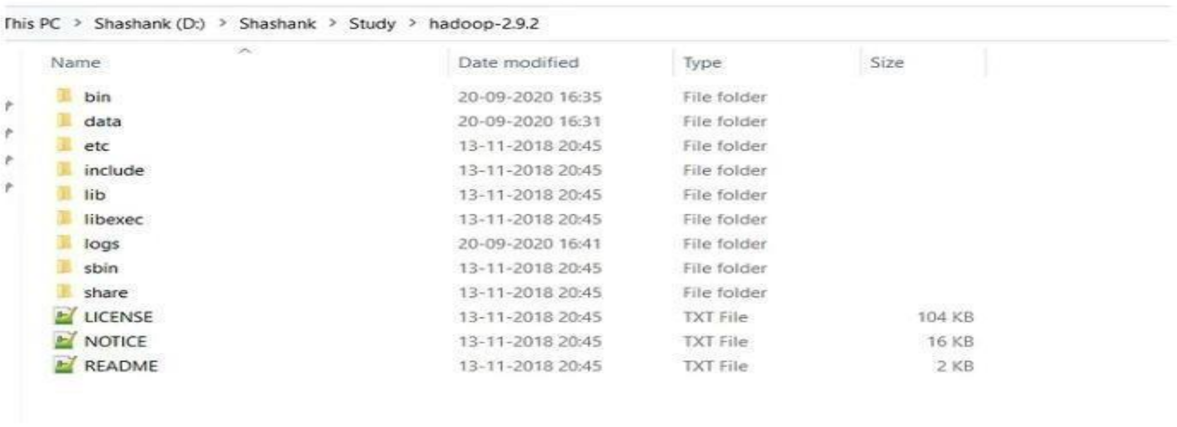
Now, once again we need to extract this tar file.



hadoop-2.9.2.tar_2	13-11-2018 23:14	TAR_2 File
hadoop-2.9.2.tar_2.gz	21-05-2020 18:56	GZ File
hadoop-3.2.1.tar	10-09-2019 22:41	TAR File
hadoop-3.2.1.tar.gz	20-05-2020 16:59	GZ File
HadoopConfiguration-FIXbin.rar	19-05-2020 13:50	RAR File
hadoop-core-0.20.2	28-05-2020 22:57	Executable Jar File
hadoop-core-1.2.1	29-06-2020 17:50	Executable Jar File
hadoop-hdfs-3.2.1 (1)	21-05-2020 16:58	Executable Jar File
hadoop-hdfs-3.2.1	21-05-2020 16:25	Executable Jar File
hadoop-mapreduce-client-core-3.3.0	12-03-2021 23:26	Executable Jar File

Fig. 3:- Extracting Hadoop Step-2

Now we can organize our Hadoop installation, we can create a folder and move the final extracted file in it. For Eg. :-



This PC > Shashank (D:) > Shashank > Study > hadoop-2.9.2

Name	Date modified	Type	Size
bin	20-09-2020 16:35	File folder	
data	20-09-2020 16:31	File folder	
etc	13-11-2018 20:45	File folder	
include	13-11-2018 20:45	File folder	
lib	13-11-2018 20:45	File folder	
libexec	13-11-2018 20:45	File folder	
logs	20-09-2020 16:41	File folder	
sbin	13-11-2018 20:45	File folder	
share	13-11-2018 20:45	File folder	
LICENSE	13-11-2018 20:45	TXT File	104 KB
NOTICE	13-11-2018 20:45	TXT File	16 KB
README	13-11-2018 20:45	TXT File	2 KB

Fig. 4:- Hadoop Directory

Please note while creating folders, DO NOT ADD SPACES IN BETWEEN THE FOLDER NAME.(it can cause issues later)
I have placed my Hadoop in D: drive you can use C: or any other drive also.

3. Setting Up Environment Variables

Another important step in setting up a work environment is to set your Systems environment variable.
To edit environment variables, go to Control Panel > System > click on the ―Advanced system settings‖ link
Alternatively, We can Right click on This PC icon and click on Properties and click on the ―Advanced system settings‖ link
Or, easiest way is to search for Environment Variable in search bar and there you GO…

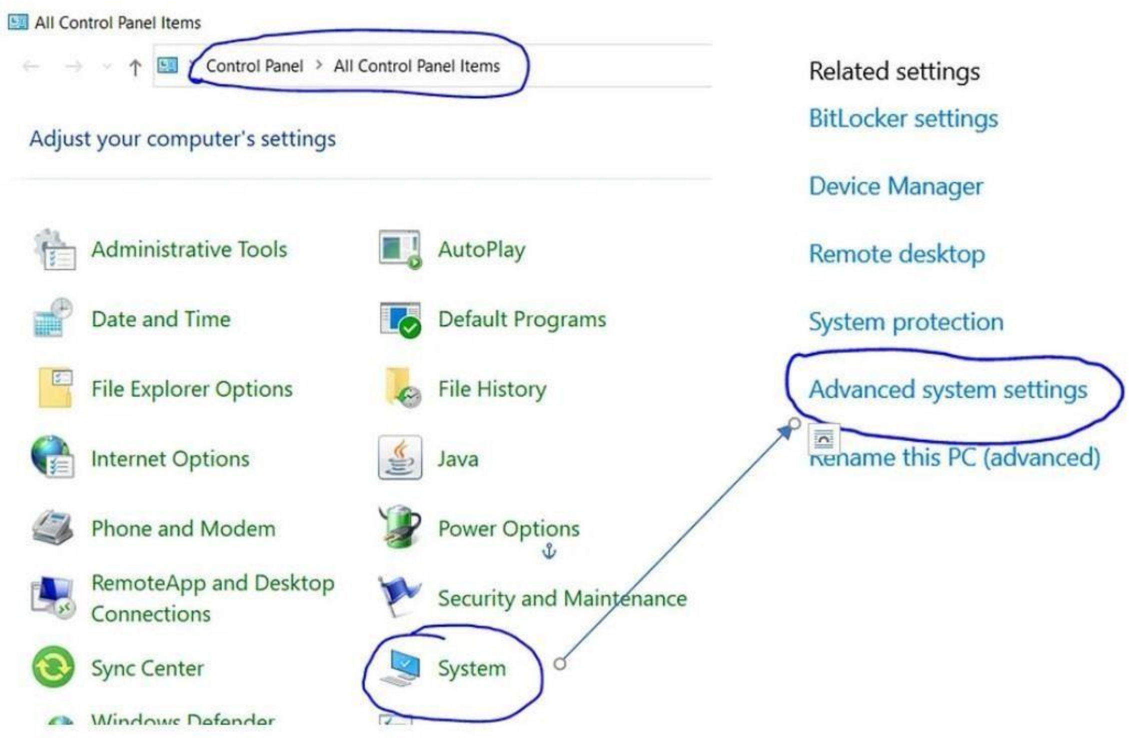
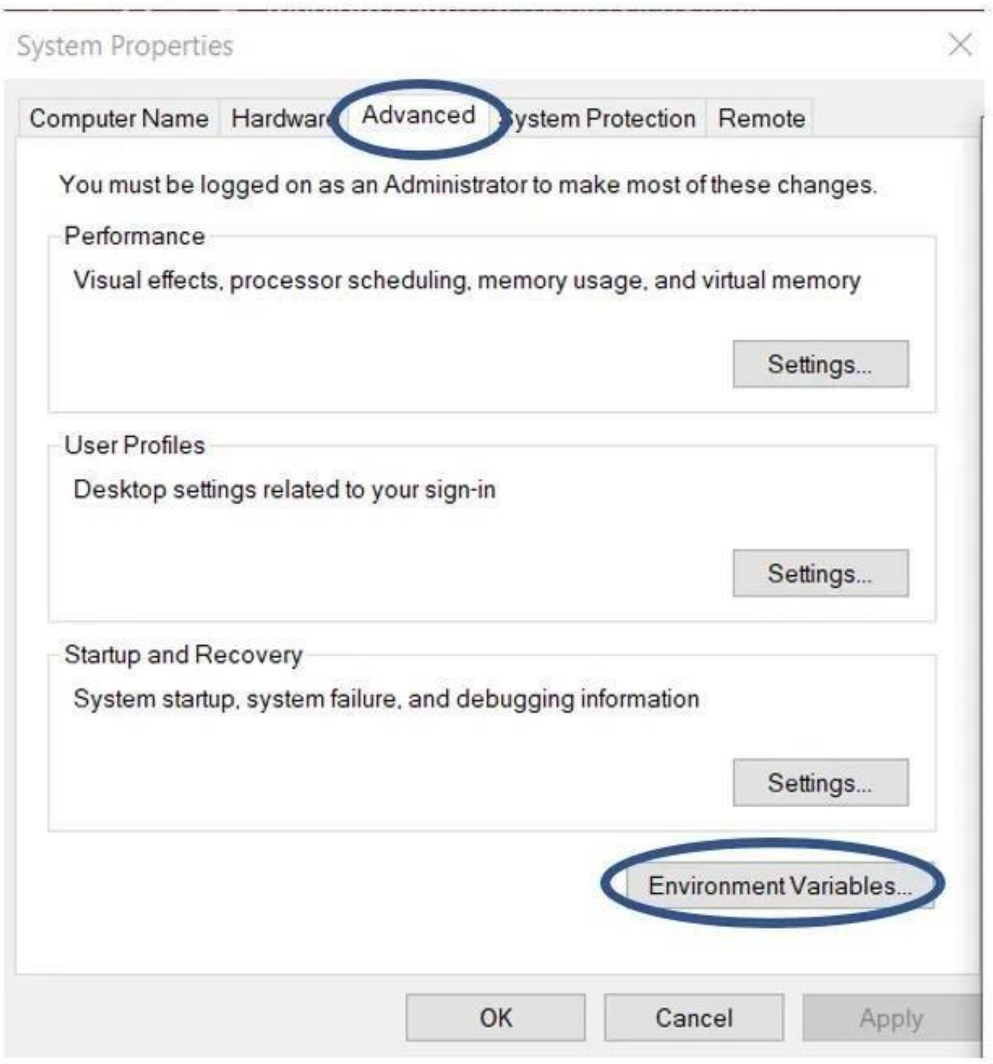


Fig. 5:- Path for Environment Variable



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

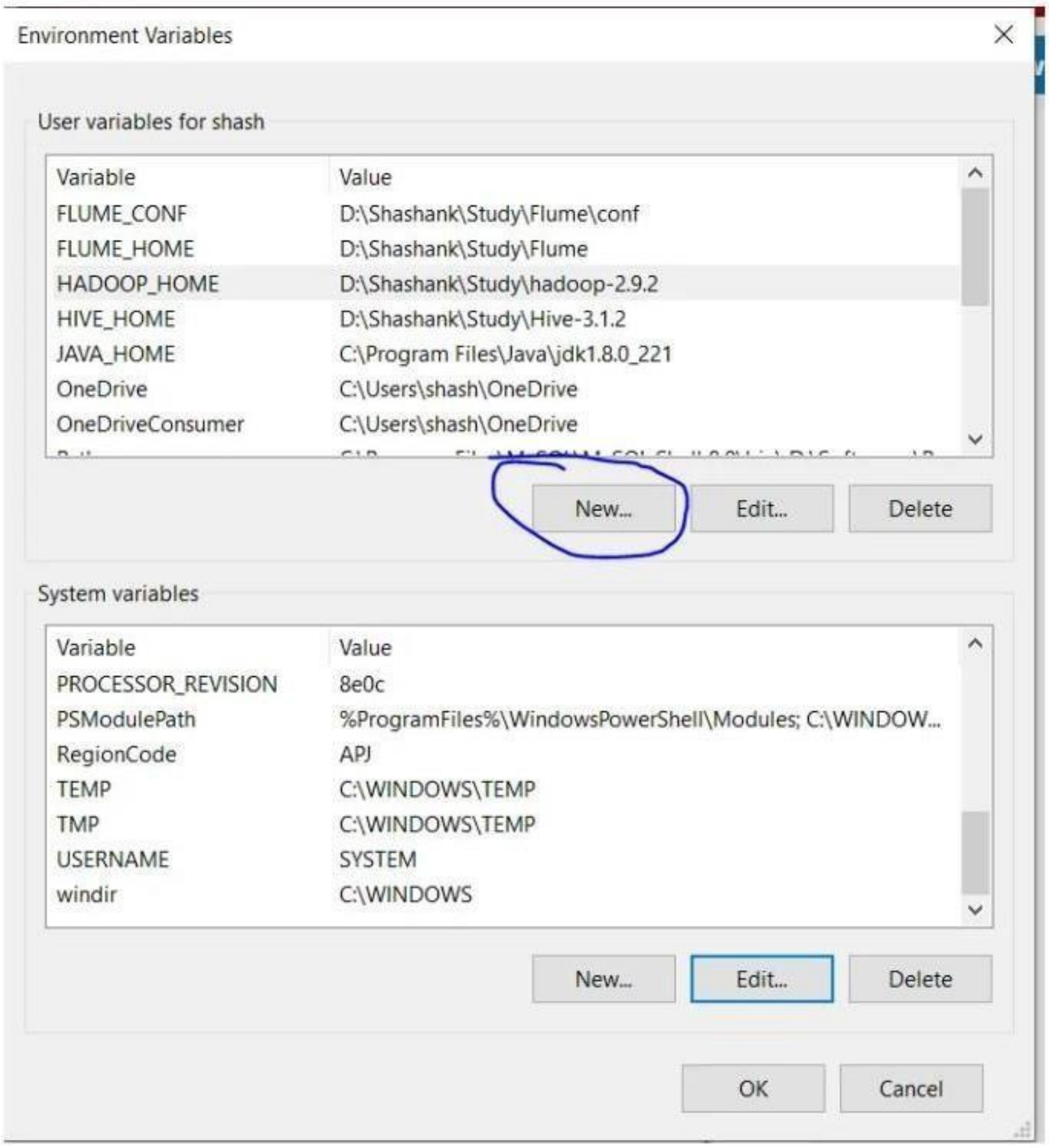
Settings...

Environment Variables...

Fig. 6:- Advanced System Settings Screen

3.1 Setting JAVA_HOME

Open environment Variable and click on ―New‖ in ―User Variable‖



Environment Variables

User variables for shash

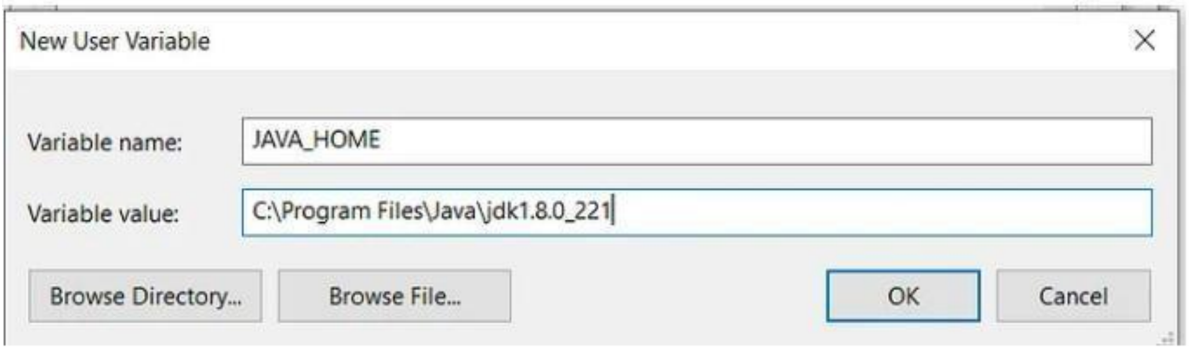
Variable	Value
FLUME_CONF	D:\Shashank\Study\Flume\conf
FLUME_HOME	D:\Shashank\Study\Flume
HADOOP_HOME	D:\Shashank\Study\hadoop-2.9.2
HIVE_HOME	D:\Shashank\Study\Hive-3.1.2
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_221
OneDrive	C:\Users\shash\OneDrive
OneDriveConsumer	C:\Users\shash\OneDrive

System variables

Variable	Value
PROCESSOR_REVISION	8e0c
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules; C:\WINDOW...
RegionCode	APJ
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
USERNAME	SYSTEM
windir	C:\WINDOWS

Fig. 7:- Adding Environment Variable

On clicking ‑New‖, we get below screen.



New User Variable

Variable name: JAVA_HOME

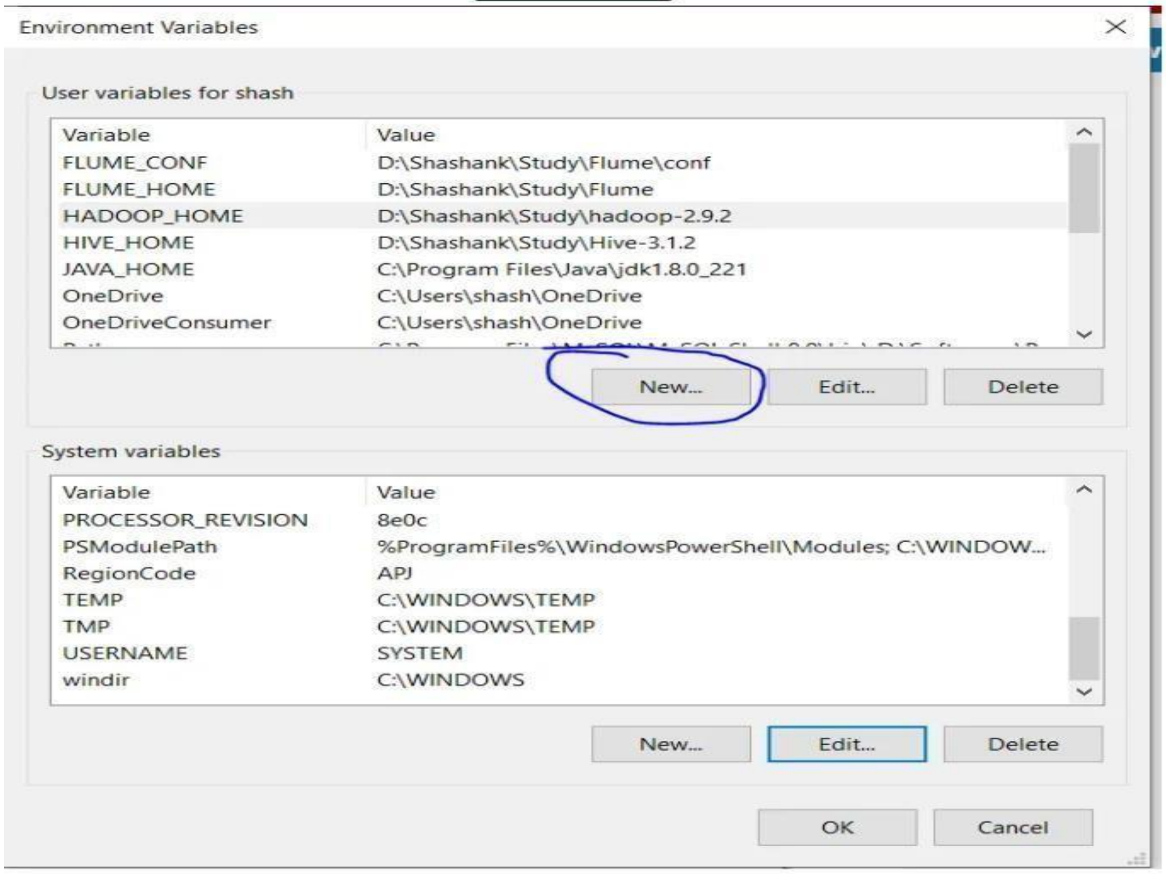
Variable value: C:\Program Files\Java\jdk1.8.0_221

Fig. 8:- Adding JAVA_HOME

Now as shown, add JAVA_HOME in variable name and path of Java(jdk) in Variable Value. Click OK and we are half done with setting JAVA_HOME.

3.2 Setting HADOOP_HOME

Open environment Variable and click on ‒New‖ in ―User Variable‖



Environment Variables

User variables for shash

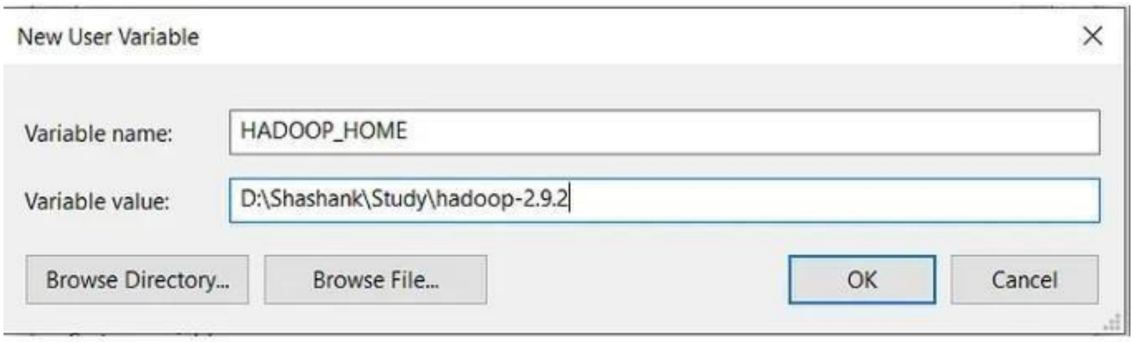
Variable	Value
FLUME_CONF	D:\Shashank\Study\Flume\conf
FLUME_HOME	D:\Shashank\Study\Flume
HADOOP_HOME	D:\Shashank\Study\hadoop-2.9.2
HIVE_HOME	D:\Shashank\Study\Hive-3.1.2
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_221
OneDrive	C:\Users\shash\OneDrive
OneDriveConsumer	C:\Users\shash\OneDrive

System variables

Variable	Value
PROCESSOR_REVISION	8e0c
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules; C:\WINDOW...
RegionCode	APJ
TEMP	C:\WINDOWS\TEMP
TMP	C:\WINDOWS\TEMP
USERNAME	SYSTEM
windir	C:\WINDOWS

Fig. 9:- Adding Environment Variable

On clicking ‒New‖, we get below screen.



New User Variable

Variable name: HADOOP_HOME

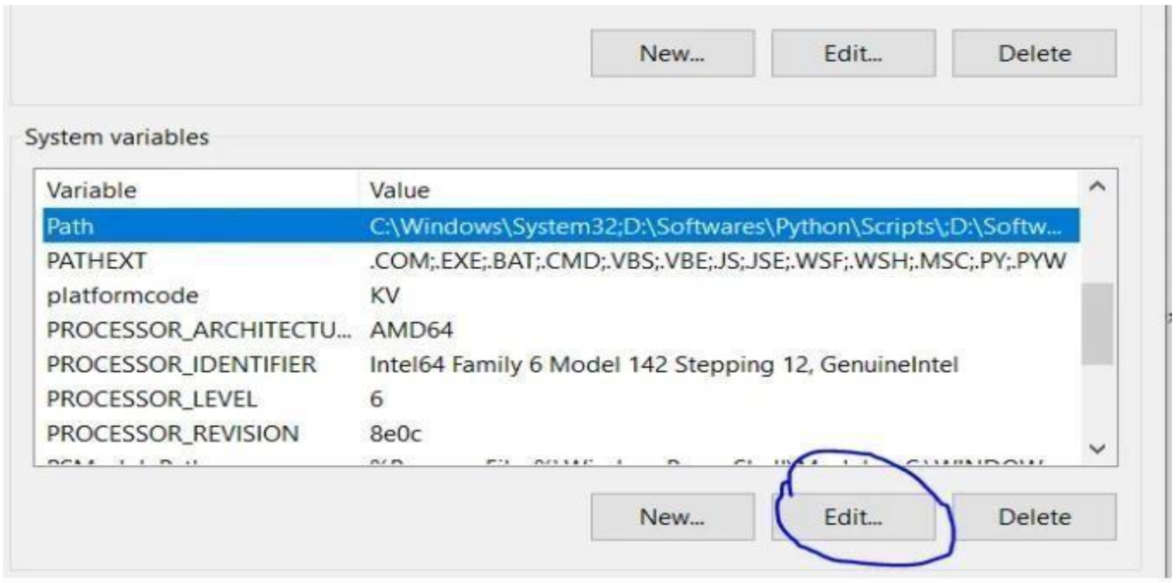
Variable value: D:\Shashank\Study\hadoop-2.9.2

Fig. 10:- Adding HADOOP_HOME

Now as shown, add HADOOP_HOME in variable name and path of Hadoop folder in Variable Value.
Click OK and we are half done with setting HADOOP_HOME.
Note:- If you want the path to be set for all users you need to select ‒New‖ from System Variables.

3.3 Setting Path Variable

Last step in setting Environment variable is setting Path in System Variable.

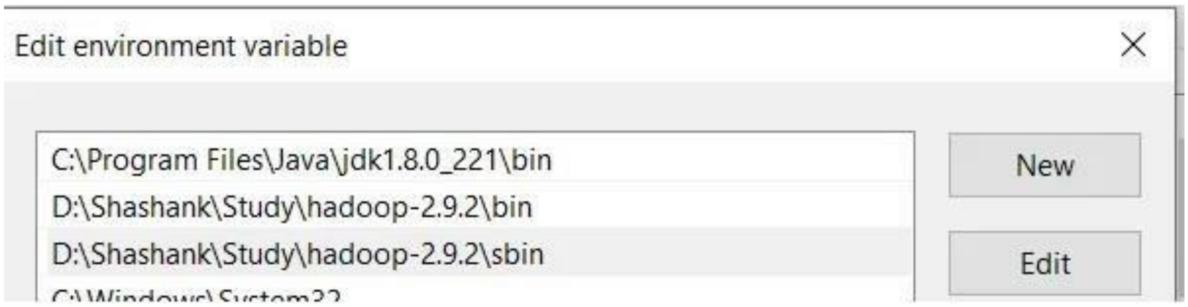


System variables

Variable	Value
Path	C:\Windows\System32;D:\Softwares\Python\Scripts\;D:\Softw...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC;.PY;.PYW
platformcode	KV
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 12, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	8e0c

Fig. 11:- Setting Path Variable

Select Path variable in the system variables and click on –Edit‖.



Edit environment variable

C:\Program Files\Java\jdk1.8.0_221\bin
D:\Shashank\Study\hadoop-2.9.2\bin
D:\Shashank\Study\hadoop-2.9.2\sbin

Fig. 12:- Adding Path

Now we need to add these paths to Path Variable one by one:-
* %JAVA_HOME%\bin
* %HADOOP_HOME%\bin
* %HADOOP_HOME%\sbin

Click OK and OK. & we are done with Setting Environment Variables.

3.4 Verify the Paths

Now we need to verify that what we have done is correct and reflecting.
Open a NEW Command Window

Run following commands

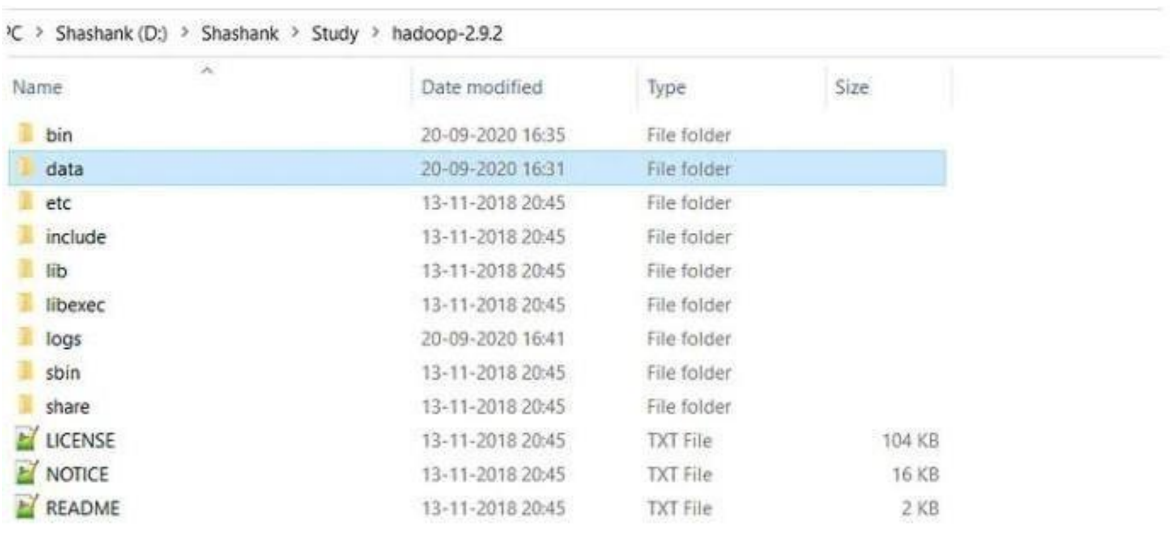
```
echo %JAVA_HOME%
echo %HADOOP_HOME%
echo %PATH%
```

4. Editing Hadoop files

Once we have configured the environment variables next step is to configure Hadoop. It has 3 parts:-

4.1 Creating Folders

We need to create a folder data in the hadoop directory, and 2 sub folders namenode and datanode

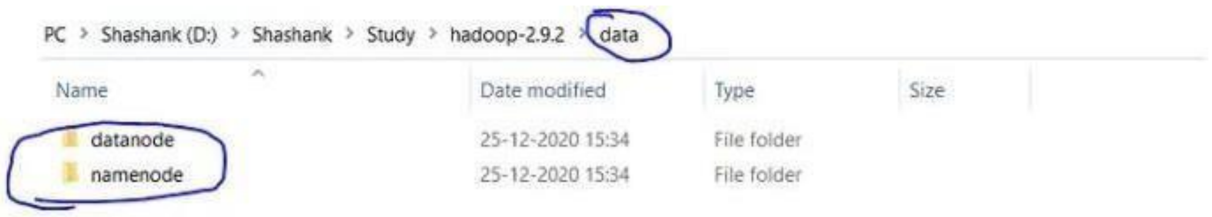


PC > Shashank (D:) > Shashank > Study > hadoop-2.9.2

Name	Date modified	Type	Size
bin	20-09-2020 16:35	File folder	
data	20-09-2020 16:31	File folder	
etc	13-11-2018 20:45	File folder	
include	13-11-2018 20:45	File folder	
lib	13-11-2018 20:45	File folder	
libexec	13-11-2018 20:45	File folder	
logs	20-09-2020 16:41	File folder	
sbin	13-11-2018 20:45	File folder	
share	13-11-2018 20:45	File folder	
LICENSE	13-11-2018 20:45	TXT File	104 KB
NOTICE	13-11-2018 20:45	TXT File	16 KB
README	13-11-2018 20:45	TXT File	2 KB

Fig. 13:- Creating Data Folder

Create DATA folder in the Hadoop directory



PC > Shashank (D:) > Shashank > Study > hadoop-2.9.2 > data

Name	Date modified	Type	Size
datanode	25-12-2020 15:34	File folder	
namenode	25-12-2020 15:34	File folder	

Fig. 14:- Creating Sub-folders

Once DATA folder is created, we need to create 2 new folders namely, namenode and datanode inside the data folder

These folders are important because files on HDFS resides inside the datanode.

4.2 Editing Configuration Files

Now we need to edit the following config files in hadoop for configuring it :-

(We can find these files in Hadoop -> etc -> hadoop)

* core-site.xml

* hdfs-site.xml

* mapred-site.xml

* yarn-site.xml

* hadoop-env.cmd

4.2.1 Editing core-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
 </property>
</configuration>
```

4.2.2 Editing hdfs-site.xml

Right click on the file, select edit and paste the following content within <configuration></configuration>tags.
Note:- Below part already has the configuration tag, we need to copy only the part inside it.
Also replace PATH~1 and PATH~2 with the path of namenode and datanode folder that we created recently(step 4.1).

```
<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
 <property>
  <name>dfs.namenode.name.dir</name>
  <value>C:\hadoop\data\namenode</value>
   </property>
 <property>
  <name>dfs.datanode.data.dir</name>
  <value>C:\hadoop\data\datanode</value>
 </property>
</configuration>
```

4.2.3 Editing mapred-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.
Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
 <property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
 </property>
</configuration>
```

4.2.4 Editing yarn-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.
Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
 <property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
 </property>
 <property>
  <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
 </property>
</configuration>
```

4.2.5 Verifying hadoop-env.cmd

Right click on the file, select edit and check if the JAVA_HOME is set correctly or not.
We can replace the JAVA_HOME variable in the file with your actual JAVA_HOME that we configured in the System Variable.

```
set JAVA_HOME=%JAVA_HOME%
         OR
set JAVA_HOME="C:\Program Files\Java\jdk1.8.0_221"
```

4.3 Replacing bin

Last step in configuring the hadoop is to download and replace the bin folder.

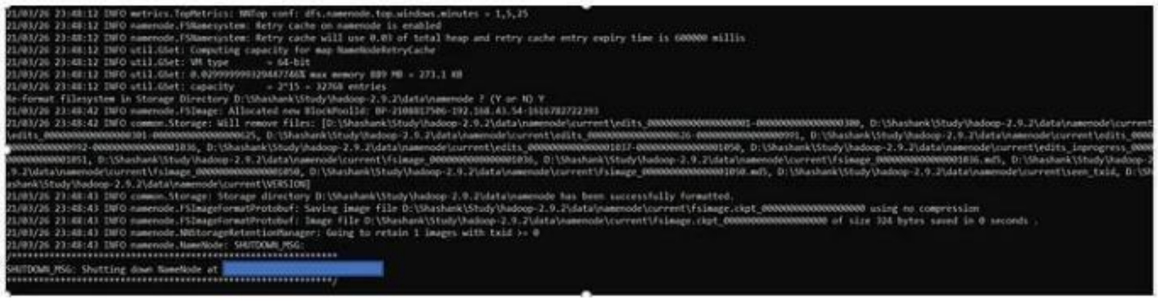
* Go to this GitHub Repo and download the bin folder as a zip.
* Extract the zip and copy all the files present under bin folder to %HADOOP_HOME%\bin
Note:- If you are using different version of Hadoop then please search for its respective bin folder and download it.

5. Testing Setup

Congratulation..!!!!!
We are done with the setting up the Hadoop in our System.
Now we need to check if everything works smoothly…

5.1 Formatting Namenode

Before starting hadoop we need to format the namenode for this we need to start a NEW Command Prompt and run below command
hadoop namenode –format



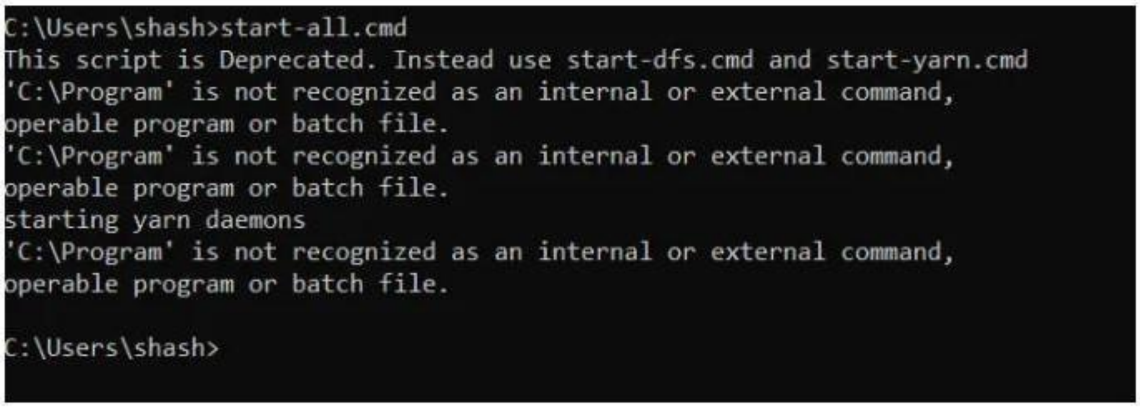
```
21/03/26 23:48:12 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
21/03/26 23:48:12 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
21/03/26 23:48:12 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
21/03/26 23:48:12 INFO util.GSet: Computing capacity for map NameNodeRetryCache
21/03/26 23:48:12 INFO util.GSet: VM type       = 64-bit
21/03/26 23:48:12 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
21/03/26 23:48:12 INFO util.GSet: capacity      = 2^15 = 32768 entries
Re-format filesystem in Storage Directory D:\Shashank\Study\hadoop-2.9.2\data\namenode ? (Y or N) Y
21/03/26 23:48:42 INFO namenode.FSImage: Allocated new BlockPoolId: BP-2108817506-192.168.43.54-1616782722393
21/03/26 23:48:42 INFO common.Storage: Will remove files: [D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\edits_0000000000000000001-0000000000000000300, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current
\edits_0000000000000000301-0000000000000000625, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\edits_0000000000000000626-0000000000000000991, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\edits_0000
0000000000000992-0000000000000001036, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\edits_0000000000000001037-0000000000000001050, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\edits_inprogress_0000
0000000000001051, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\fsimage_0000000000000001036, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\fsimage_0000000000000001036.md5, D:\Shashank\Study\hadoop-2
.9.2\data\namenode\current\fsimage_0000000000000001050, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\fsimage_0000000000000001050.md5, D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\seen_txid, D:\Sh
ashank\Study\hadoop-2.9.2\data\namenode\current\VERSION]
21/03/26 23:48:43 INFO common.Storage: Storage directory D:\Shashank\Study\hadoop-2.9.2\data\namenode has been successfully formatted.
21/03/26 23:48:43 INFO namenode.FSImageFormatProtobuf: Saving image file D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\fsimage.ckpt_0000000000000000000 using no compression
21/03/26 23:48:43 INFO namenode.FSImageFormatProtobuf: Image file D:\Shashank\Study\hadoop-2.9.2\data\namenode\current\fsimage.ckpt_0000000000000000000 of size 324 bytes saved in 0 seconds .
21/03/26 23:48:43 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
21/03/26 23:48:43 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at [redacted]
************************************************************/
```

Fig. 15:- Formatting Namenode

Note:- This command formats all the data in namenode. So, its advisable to use only at the start and do not use it every time while starting hadoop cluster to avoid data loss.

5.2 Launching Hadoop

Now we need to start a new Command Prompt remember to run it as administrator to avoid permission issues and execute below commands
start-all.cmd

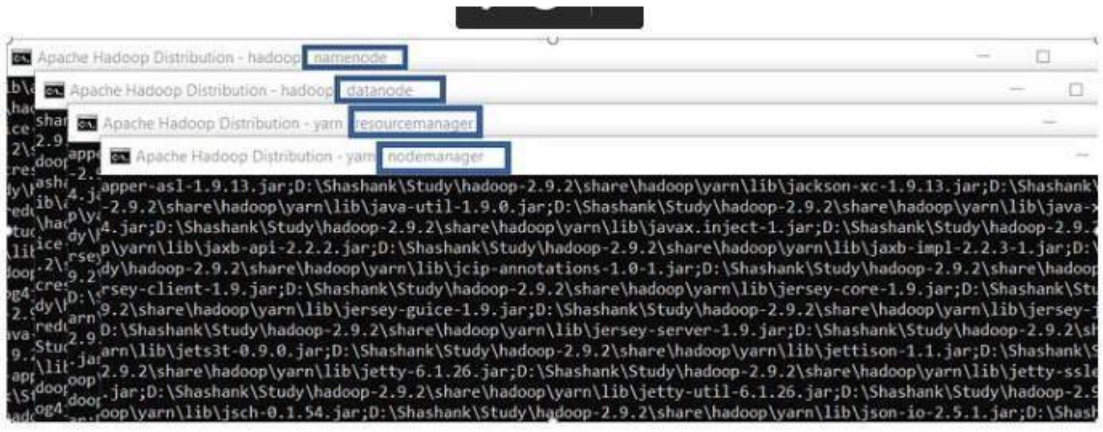


```
C:\Users\shash>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
starting yarn daemons
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\shash>
```

Fig. 16:- start-all.cmd

This will open 4 new cmd windows running 4 different Daemons of hadoop:-
* Namenode
* Datanode
* Resourcemanager
* Nodemanager



```
Apache Hadoop Distribution - hadoop namenode
Apache Hadoop Distribution - hadoop datanode
Apache Hadoop Distribution - yarn resourcemanager
Apache Hadoop Distribution - yarn nodemanager
apper-asl-1.9.13.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jackson-xc-1.9.13.jar;D:\Shashank\
-2.9.2\share\hadoop\yarn\lib\java-util-1.9.0.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\java-x
4.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\javax.inject-1.jar;D:\Shashank\Study\hadoop-2.9.
p\yarn\lib\jaxb-api-2.2.2.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jaxb-impl-2.2.3-1.jar;D:\
dy\hadoop-2.9.2\share\hadoop\yarn\lib\jcip-annotations-1.0-1.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop
rsey-client-1.9.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jersey-core-1.9.jar;D:\Shashank\Stu
9.2\share\hadoop\yarn\lib\jersey-guice-1.9.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jersey-
D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jersey-server-1.9.jar;D:\Shashank\Study\hadoop-2.9.2\sh
arn\lib\jets3t-0.9.0.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jettison-1.1.jar;D:\Shashank\S
2.9.2\share\hadoop\yarn\lib\jetty-6.1.26.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jetty-ssl
.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\jetty-util-6.1.26.jar;D:\Shashank\Study\hadoop-2.9
oop\yarn\lib\jsch-0.1.54.jar;D:\Shashank\Study\hadoop-2.9.2\share\hadoop\yarn\lib\json-io-2.5.1.jar;D:\Shas
```

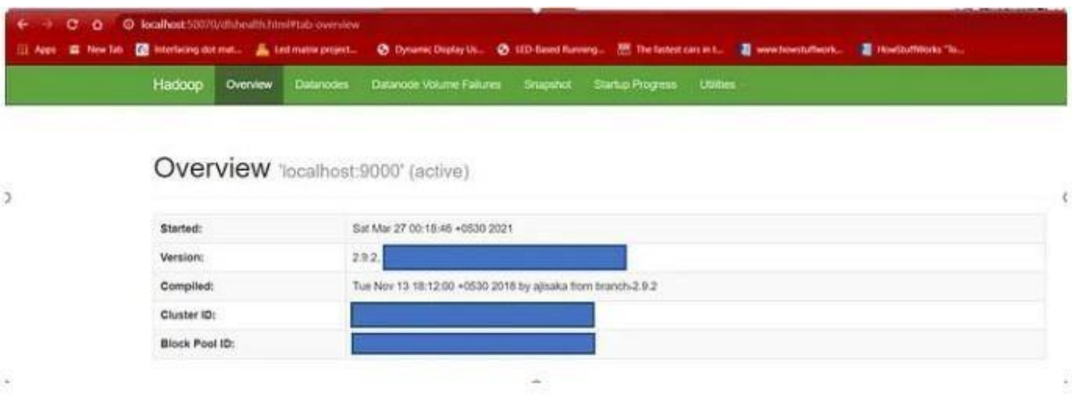
Fig. 17:- Hadoop Deamons

Note:- We can verify if all the daemons are up and running using jps command in new cmd window.

6. Running Hadoop (Verifying Web UIs)

6.1 Namenode

Open localhost:50070 in a browser tab to verify namenode health.



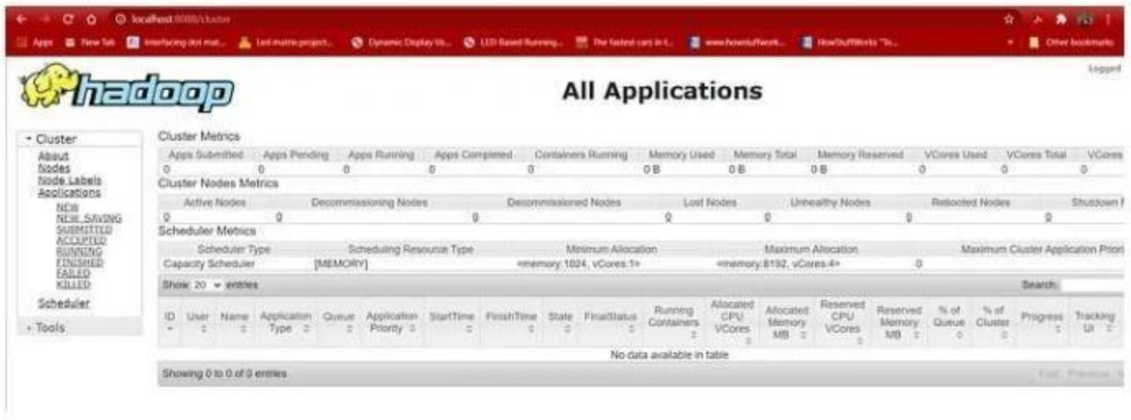
Overview 'localhost:9000' (active)

Started:	Sat Mar 27 00:18:46 +0530 2021
Version:	2.9.2, [redacted]
Compiled:	Tue Nov 13 18:12:00 +0530 2018 by ajisaka from branch-2.9.2
Cluster ID:	[redacted]
Block Pool ID:	[redacted]

Fig. 18:- Namenode Web UI

6.2 Resourcemanger

Open localhost:8088 in a browser tab to check resourcemanager details.



All Applications

Cluster Metrics

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

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown
0	0	0	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priori
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries Search:

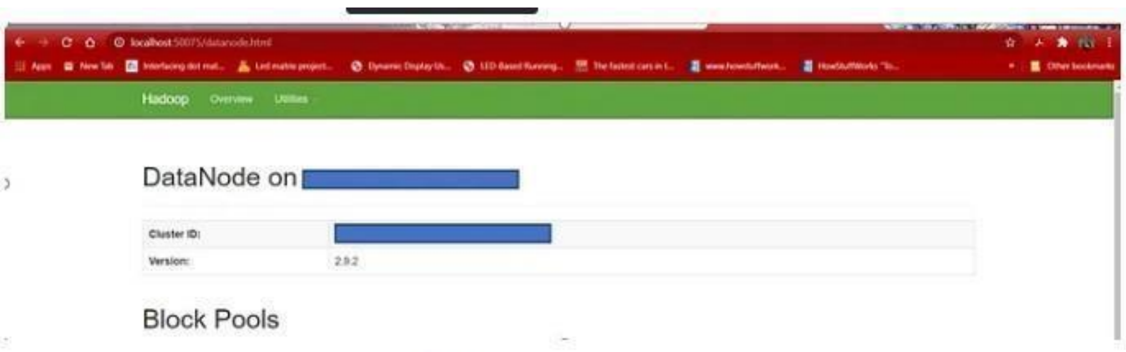
ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI
No data available in table																		

Showing 0 to 0 of 0 entries

Fig. 19:- Resourcemanager Web UI

6.3 Datanode

Open localhost:50075 in a browser tab to checkout datanode.



DataNode on [redacted]

Cluster ID:	[redacted]
Version:	2.9.2

Block Pools

Fig. 20:- Datanode Web UI

OUTPUT:

EXPERIMENT:

Implement Word Count/ Frequency Programs Using Map Reduce.

PROGRAM:

AIM**:** To count a given number using map reduce functions.

Hadoop Streaming API for helping us passing data between our Map and Reduce code via STDIN (standard input) and STDOUT (standard output).

Note : Change the file has execution permission (chmod +x /home/hduser/mapper.py)

Change the file has execution permission (chmod +x /home/hduser/reducer.py

Mapper program

mapper.py

```
import sys
# input comes from STDIN (standard input)
for line in sys.stdin:
      line = line.strip() # remove leading and trailing whitespace
   words = line.split()# split the line into words
   # increase counters
   for word in words:
      # write the results to STDOUT (standard output);
      # what we output here will be the input for the
      # Reduce step, i.e. the input for reducer.py
      # tab-delimited; the trivial word count is 1
      print '%s\t%s' % (word, 1)
Reducer program
"""reducer.py"""
from operator import itemgetter
import sys
current_word = None
current_count = 0
word = None

# input comes from STDIN
for line in sys.stdin:
   line = line.strip() # remove leading and trailing whitespace
   # parse the input we got from mapper.py
   word, count = line.split('\t', 1)
   # convert count (currently a string) to int
   try:
      count = int(count)
   except ValueError:
      # count was not a number, so silently
      # ignore/discard this line
      continue

   # this IF-switch only works because Hadoop sorts map output
   # by key (here: word) before it is passed to the reducer
   if current_word == word:
      current_count += count
   else:
        if current_word:
            # write result to STDOUT
            print '%s\t%s' % (current_word, current_count)
        current_count = count
        current_word = word

# do not forget to output the last word if needed!
if current_word == word:
    print '%s\t%s' % (current_word, current_count)
```

Test the code (cat data | map | sort | reduce)

```
hduser@ubuntu:~$ echo "foo foo quux labs foo bar quux" | /home/hduser/mapper.py
foo     1
foo     1
quux    1
labs    1
foo     1
bar     1
quux    1

hduser@ubuntu:~$ echo "foo foo quux labs foo bar quux" | /home/hduser/mapper.py | sort -k1,1 | /home/hduser/reducer.py
bar     1
foo     3
labs    1
quux    2

hduser@ubuntu:~$ cat /tmp/gutenberg/20417-8.txt | /home/hduser/mapper.py
 The     1
 Project 1
 Gutenberg       1
 EBook  1
 of      1
```

OUTPUT: