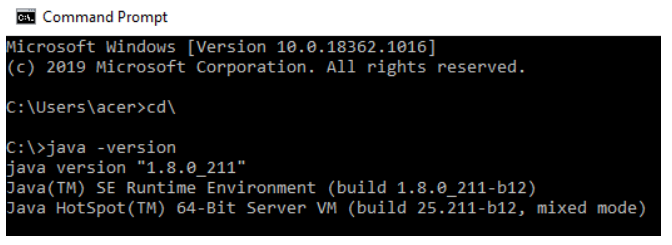
Practical-8

|  |  |
| --- | --- |
| **TASK :** | **Hadoop installation on Windows 10 ( Single Node/Cluster)** |

**Prerequisite :**

To install Hadoop you should have Java version 1.8 or greater in your system. 

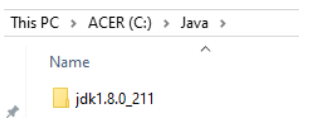
If java is not installed in your system, then go to the following link:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Keep the java folder directly under the local disk directory

(C:\Java\jdk1.8.0\_211) rather than in Program Files (C:\Program

Files\Java\jdk1.8.0\_211) as it can create errors afterwards



Accept the license and Download the file according to your operating system.

Download Hadoop:

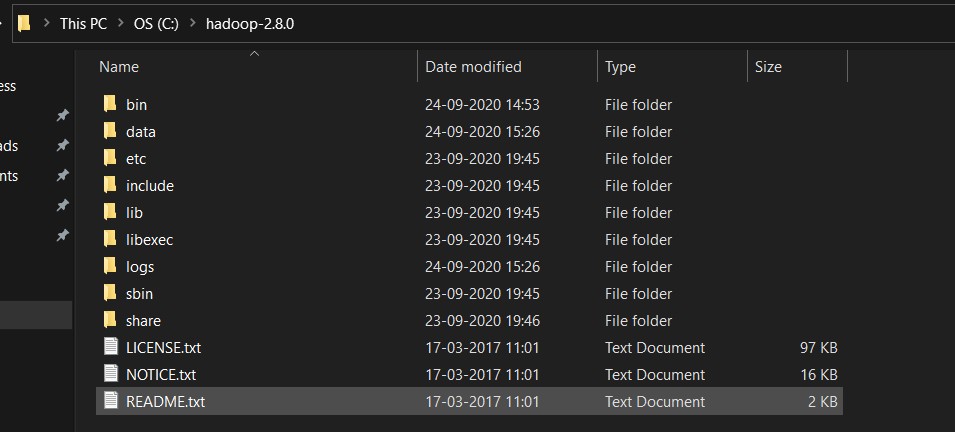
* After downloading java version 1.8, download Hadoop-2.8.0 from this link:

<http://archive.apache.org/dist/hadoop/core//hadoop-2.8.0/hadoop-2.8.0.tar.gz> .

OR

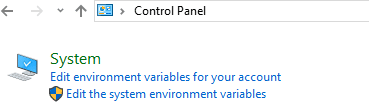
<http://archive.apache.org/dist/hadoop/core/hadoop-2.8.0/hadoop-2.8.0.tar.gz>

Extract it to a folder.

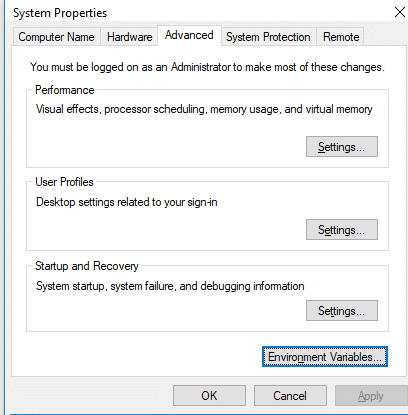


**Setup System Environment Variables**

* Open control panel to edit the system environment variable.



* Go to environment variable in system properties.

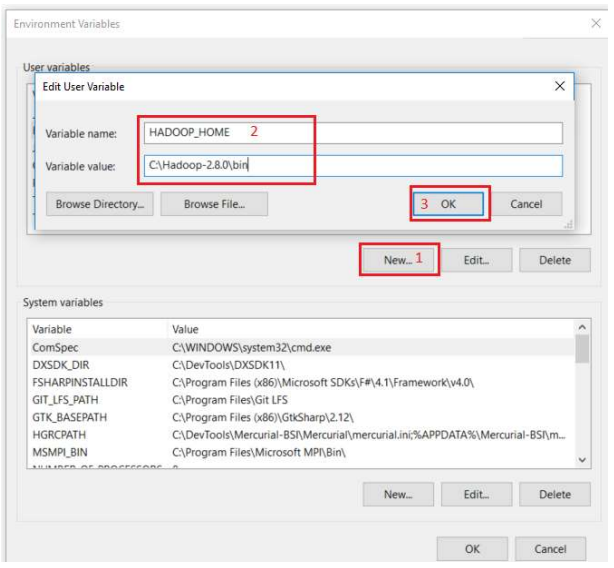


Create a new user variable. Put the Variable\_name as HADOOP\_HOME and

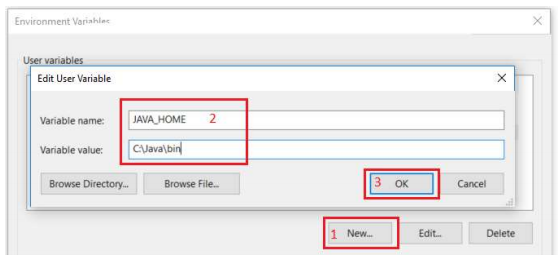
Variable\_value as the path of the bin folder where you extracted hadoop. (Set

the path HADOOP\_HOME Environment variable on windows 10(see Step 1,2,3

and 4 below).



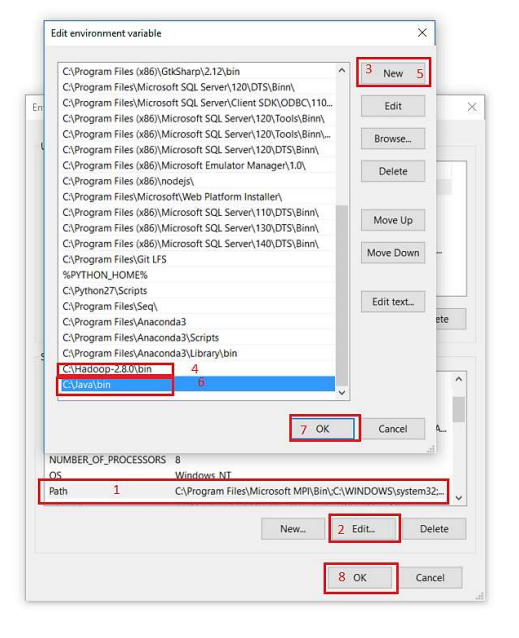
Likewise, create a new user variable with variable name as JAVA\_HOME and variable value as the path of the bin folder in the Java directory. (Set the path JAVA\_HOME Environment variable on windows 10(see Step 1, 2, 3 and 4).



Now we need to set Hadoop bin directory and Java bin directory path in user variable path.

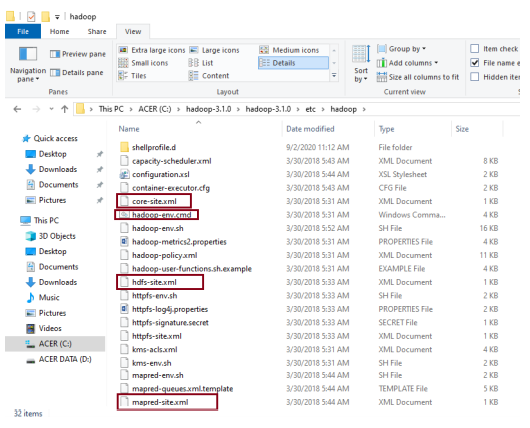
Edit Path in user variable.

(If you wish to configure multi-clusters node then add path in system variable)



**Hadoop Configuration**

* Now we need to edit some files located in the Hadoop directory of the etc folder where we installed Hadoop. The files that need to be edited have been highlighted.



* Edit file C:\hadoop-2.8.0\etc\hadoop\core-site.xml, paste below xml paragraph and save this file4

<configuration>

<property>

<name>**fs.defaultFS**</name>

<value>**hdfs://localhost:9000**</value>

</property>

</configuration>

* Rename “mapred-site.xml.template” to “mapred-site.xml” and edit this file **C:/Hadoop-2.8.0/etc/hadoop/mapred-site.xml**, paste below xml paragraph and save this file.

<configuration>

<property>

<name>**mapreduce.framework.name**</name>

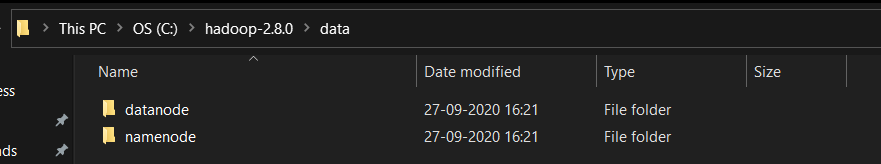
<value>**yarn**</value>

</property>

</configuration>

Create folder **“data”** under **“C:\Users\User\hadoop-2.8.0.tar\hadoop-2.8.0\data”**

* Create folder **“datanode”** under **“C:\Users\User\hadoop-2.8.0.tar\hadoop-2.8.0\data\datanode”**
* Create folder **“namenode”** under **“C:\Users\User\hadoop-2.8.0.tar\hadoop-2.8.0\data\namenode”**



* Edit file **C:\Users\User\hadoop-2.8.0.tar\hadoop-2.8.0\etc\hadoop\hdfs-site.xml**, paste below xml paragraph and save this file.

<configuration>

<property>

<name>**dfs.replication**</name>

<value>**1**</value>

</property>

<property>

<name>**dfs.namenode.name.dir**</name>

<value>**/hadoop-2.8.0/data/namenode**</value>

</property>

<property>

<name>**dfs.namenode.data.dir**</name>

<value**>/hadoop-2.8.0/data/datanode**</value>

</property>

</configuration>

* Edit file **C:\Users\User\hadoop-2.8.0.tar\hadoop-2.8.0\etc\hadoop\yarn-site.xml**, paste below xml paragraph and save this file.

<configuration>

<property>

<name>**yarn.nodemanager.aux-services**</name>

<value>**mapreduce\_shuffle**</value>

</property>

<property>

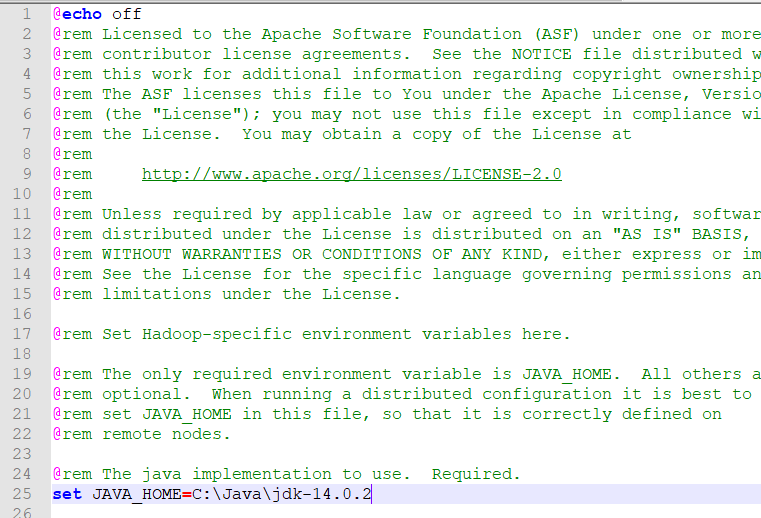
<name>**yarn.nodemanager.auxservices.mapreduce.shuffle.class**</name>

<value>**org.apache.hadoop.mapred.ShuffleHandler**</value>

</property>

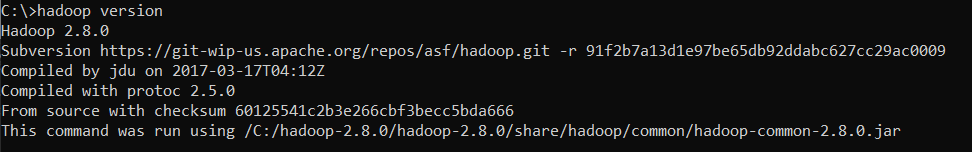
</configuration>

* Edit file **C:/Hadoop-2.8.0/etc/Hadoop/Hadoop-env.** cmd by closing the command line **“JAVA\_HOME=%JAVA\_HOME%”** instead of set **“JAVA\_HOME=C:\Java\jdk14.0.2”**

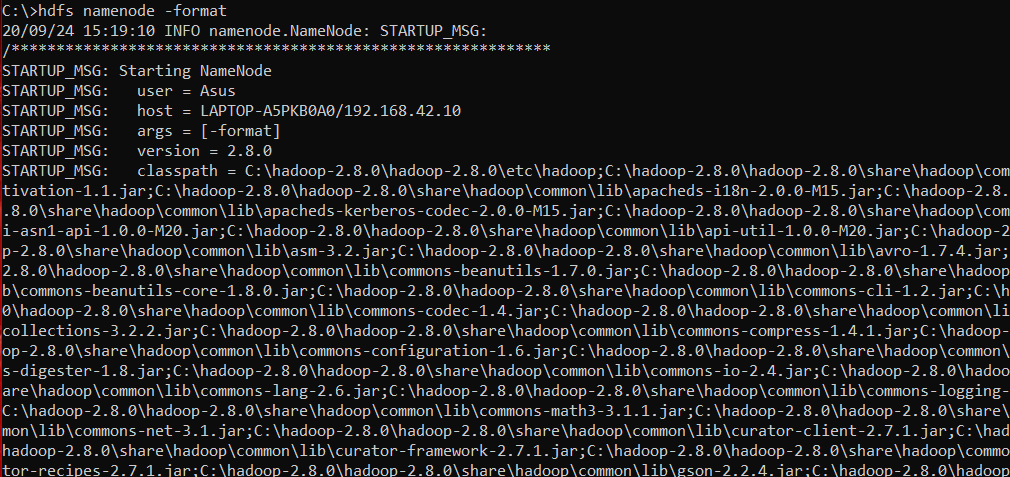


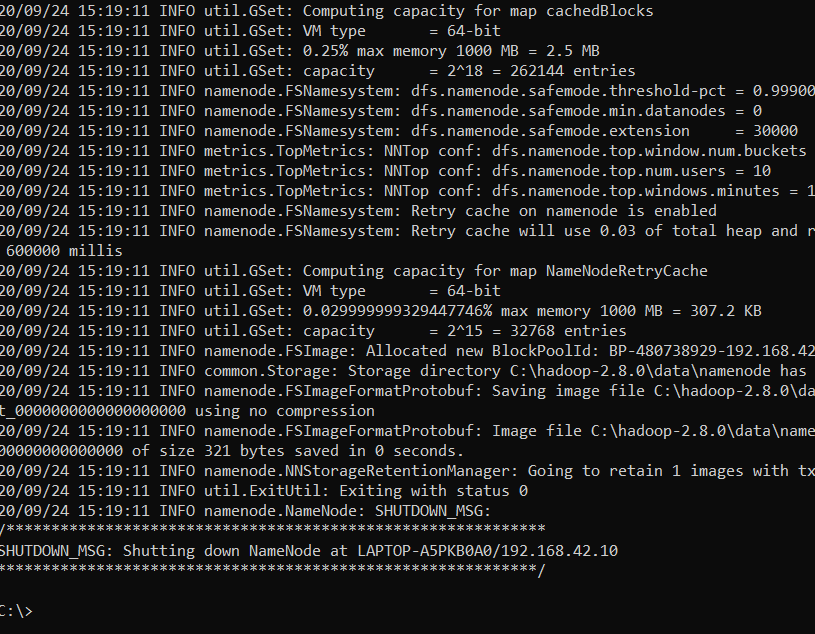
* Hadoop needs windows OS specific files which does not come with default download of Hadoop. To include this files, replace the bin folder in Hadoop directory with the bin provided in this link.
* Download file: [https://github.com/MuhammadBilalYar/Hadoop-On- Window/blob/master/Hadoop%20Configuration.zip](https://github.com/MuhammadBilalYar/Hadoop-On-Window/blob/master/Hadoop%20Configuration.zip)
* Delete file bin on C:\Hadoop-2.8.0\bin, replaced by file on file just downloaded (from Hadoop Configuration.zip). Check whether Hadoop is successfully installed by running this command on cmd

C:\>hadoop version



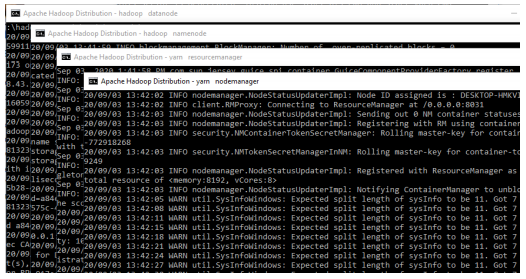
* Since it doesn’t throw error and successfully shows the Hadoop version means Hadoop is successfully installed in the system.
* Open cmd and type command **“hdfs namenode -format”.** You will see .





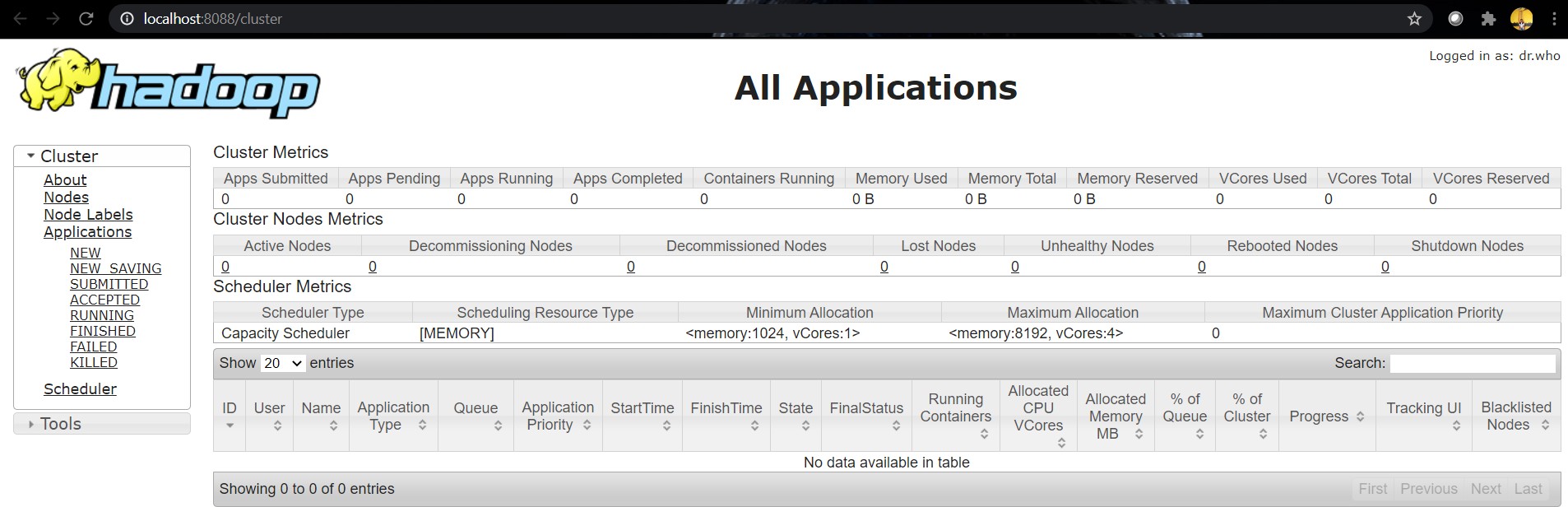
**Testing**

Open cmd and change directory to **“C:\Hadoop-2.8.0\sbin”** and type **“start-all.cmd”** to start apache.

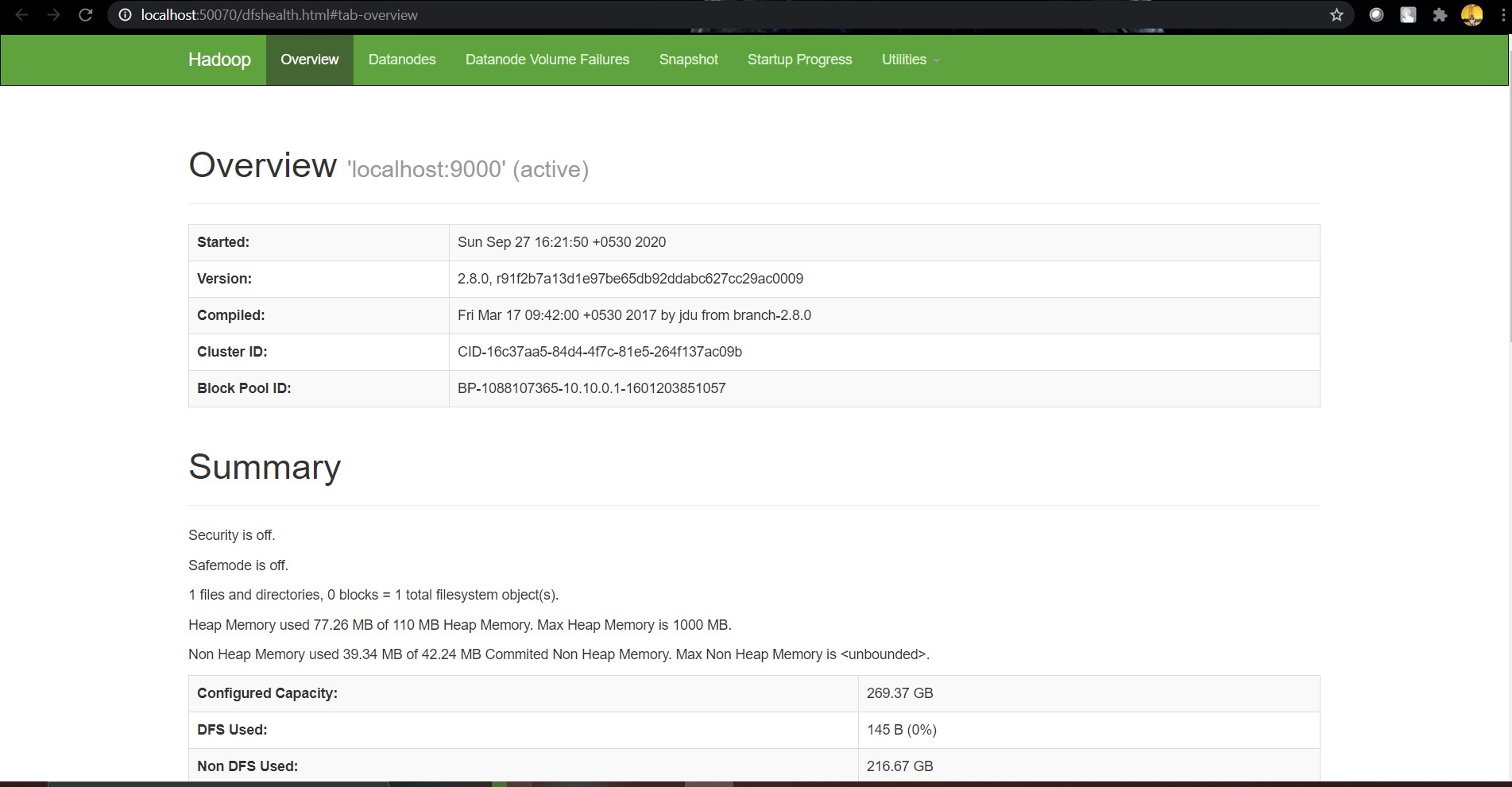


Make sure these apps are running

* + Hadoop Namenode
  + Hadoop Datanode
  + YARN Resource Manager
  + YARN Node Manager

Open: <http://localhost:8080>

Open: <http://localhost:50070>



**Congratulations, Hadoop installed.**

Working with HDFS

* Open command prompt and change the directory **“C:\hadoop-2.8.0\sbin”.**
* Create a directory named “**sample**” in my Hadoop directory using the following command.

**hdfs dfs –mkdir /sample**

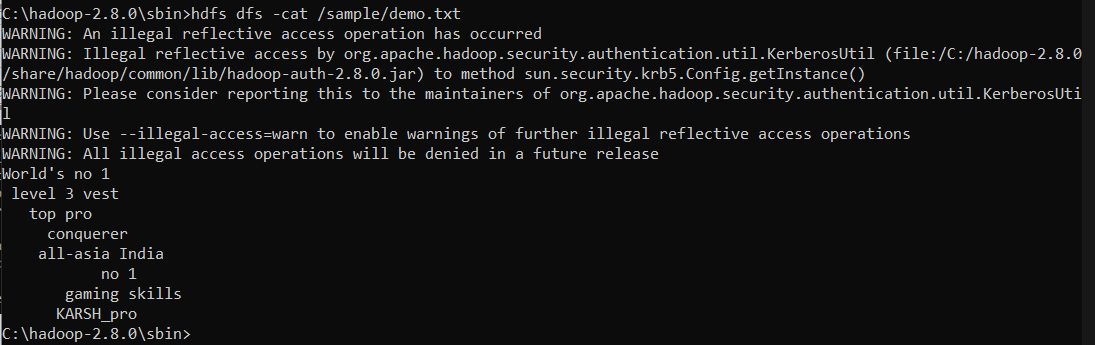
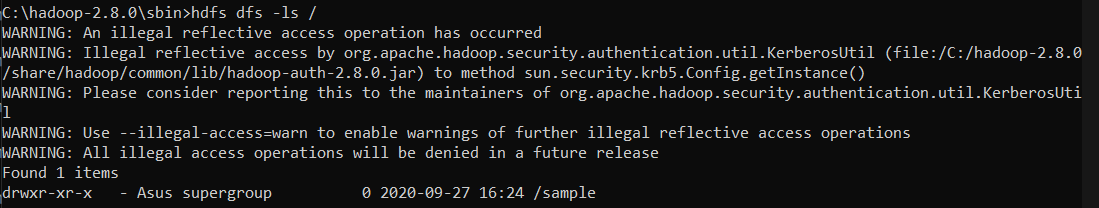
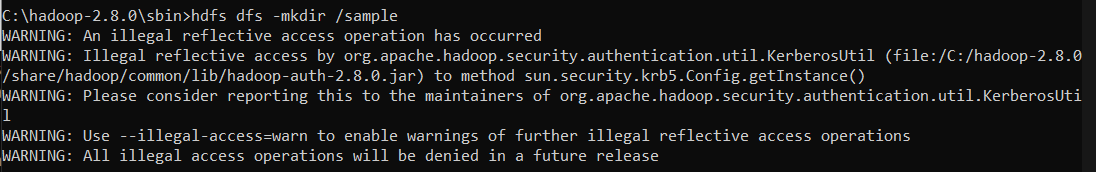
* To verify if the directory is created in hdfs, we will use ‘**ls’** command which will list the files present in hdfs-

**hdfs dfs –ls /**

* Copy a text file named “**hello.txt**” from my local file system to this folder that we have just created in hdfs using copyFromLocal command.

**hdfs dfs –copyFromLocal C:\demo.txt /sample**

* To verify if the file is copied to the folder, we will use ‘ls’ command by specifying the folder name which will read the list of files in that folder

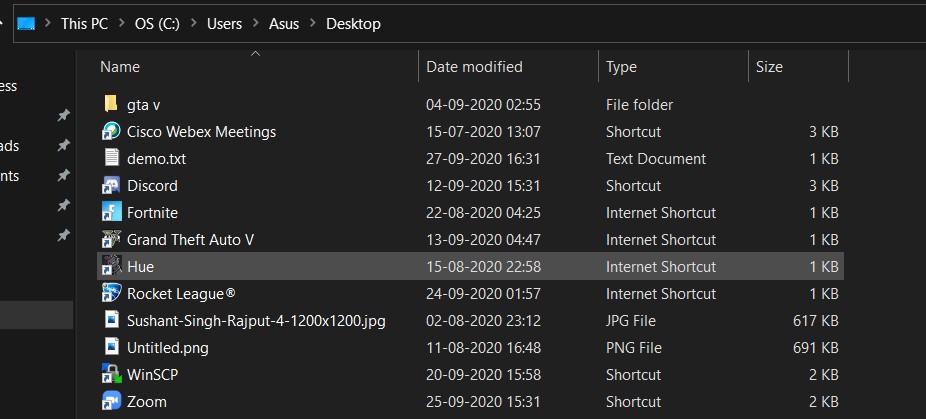
**hdfs dfs –ls /sample**

* To view the contents of the file we copied, we will use cat command

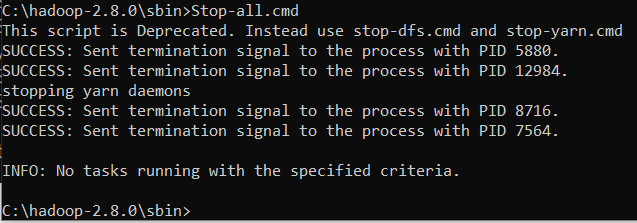
**hdfs dfs –cat /sample/demo.txt**

* To copy file from hdfs to local directory, we will use get command

**hdfs dfs –get /sample/demo.txt C:\Users\Asus\Desktop\**



* If you wish to stop Hadoop then command

**Stop-all.cmd**

**TASK :- Perform MapReduce Program in CloudX-Lab.**

* Mappers and Reducers in Hadoop are written to get their inputs from stdin, and output their tuples to stdout. (Word Count Program)

**Mapper.py:**

#!/usr/bin/env python

import sys

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# split the line into words

words = line.split()

# 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.py:**

#!/usr/bin/env python

from operator import itemgetter

import sys

current\_word = None

current\_count = 0

word = None

# input comes from STDIN

for line in sys.stdin:

# remove leading and trailing whitespace

line = line.strip()

# 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:

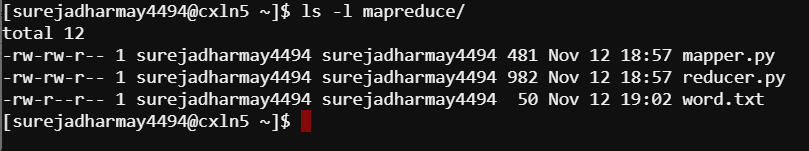
print '%s\t%s' % (current\_word, current\_count)

**word.txt:**

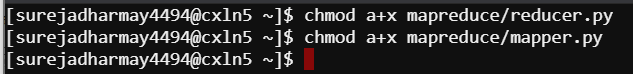
Cat mouse lion deer Tiger lion Elephant lion deer

**STEPS :**

1. Copy the manner.py and reducer.py in local webconsole(cloudXLab) folder .

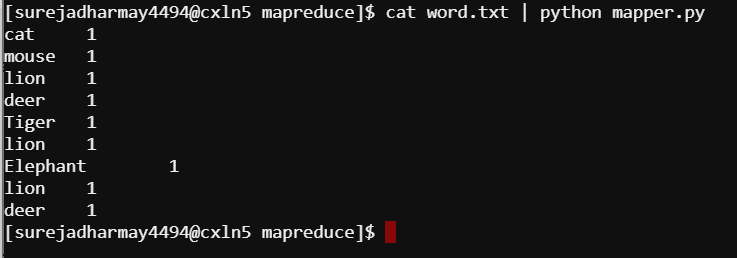


1. Change the permission of both files using :

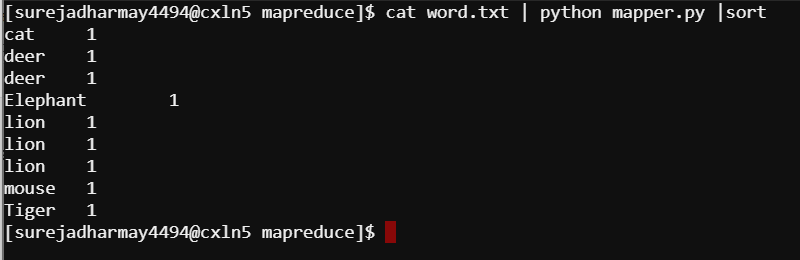


1. Perform following command to run map reducer program .

First :



Second :



Third :

