

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**WORK INTEGRATED LEARNING PROGRAMMES**

**ASSIGNMENT**

**MapReduce Algorithm Based Word-Count Program**

**Submitted By-**

**Name: Komal Nitesh Navale**

**Email:** [**2020mt93614@wilp.bits-pilani.ac.in**](mailto:2020mt93614@wilp.bits-pilani.ac.in)

**ID: 2020mt93614**

**Assignment Question:**

2. Write a MapReduce based program to display/generate a word count for each unique keyword of any given text document. This program can be implemented either locally on your system or on a virtual machine created using AWS EC2 or Google Cloud Platform’s Compute Engine. [2.5 Marks]

* **I have solved this question in 2 ways**

**First Way: By creating the Microsoft Azure HDInsight Hadoop Cluster and then using the YARN to run Program in Python**

Contents

|  |  |  |
| --- | --- | --- |
| Sr. No. | Step Name | Page No. |
| 1 | **Prerequisite Softwares to be installed in Windows 10** | 3 |
| 2 | **Development of MapReduce Word Count Program in Python** | 3 |
| 3 | **Installation of OpenSSH Client & Server** | 4 |
| 4 | **Creation of Azure HDInsight Hadoop Cluster** | 7 |
| 5 | **Program Execution** | 9 |
| 6 | **Program Output** | 13 |

**OR**

**Second Way: By installing Hadoop & Java on Windows 10 and then using in-build hadoop-mapreduce-examples.jar**

Contents

|  |  |  |
| --- | --- | --- |
| Sr. No. | Step Name | Page No. |
| 1 | **Prerequisite softwares to be installed in Windows 10** | 14 |
| 2 | **Installation of 7 Zip** | 14 |
| 3 | **Installation and Configuration of JAVA**  3.1 Installation  3.2 Windows Configuration | 15  15  15 |
| 4 | **Installation and Configuration of Hadoop**  4.1 Installation  4.2 Hadoop configuration  4.3 Windows configuration | 18  18  19  22 |
| 5 | **Program Execution** | 25 |
| 6 | **Program Output** | 30 |

**First Way: By creating the Microsoft Azure HDInsight Hadoop Cluster and then using the YARN to run Program in Python**

# Prerequisite softwares to be installed in Windows 10

## Microsoft Azure Account

## PyCharm Community Edition

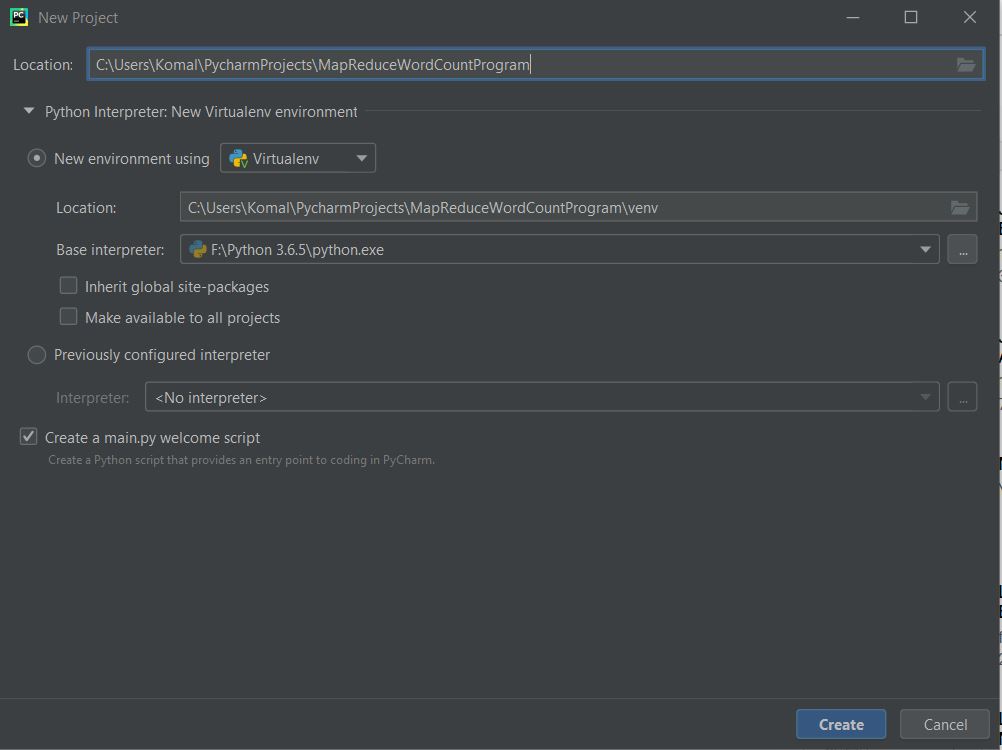
## OpenSSH Client & Server

## Windows Powershell and Command Prompt

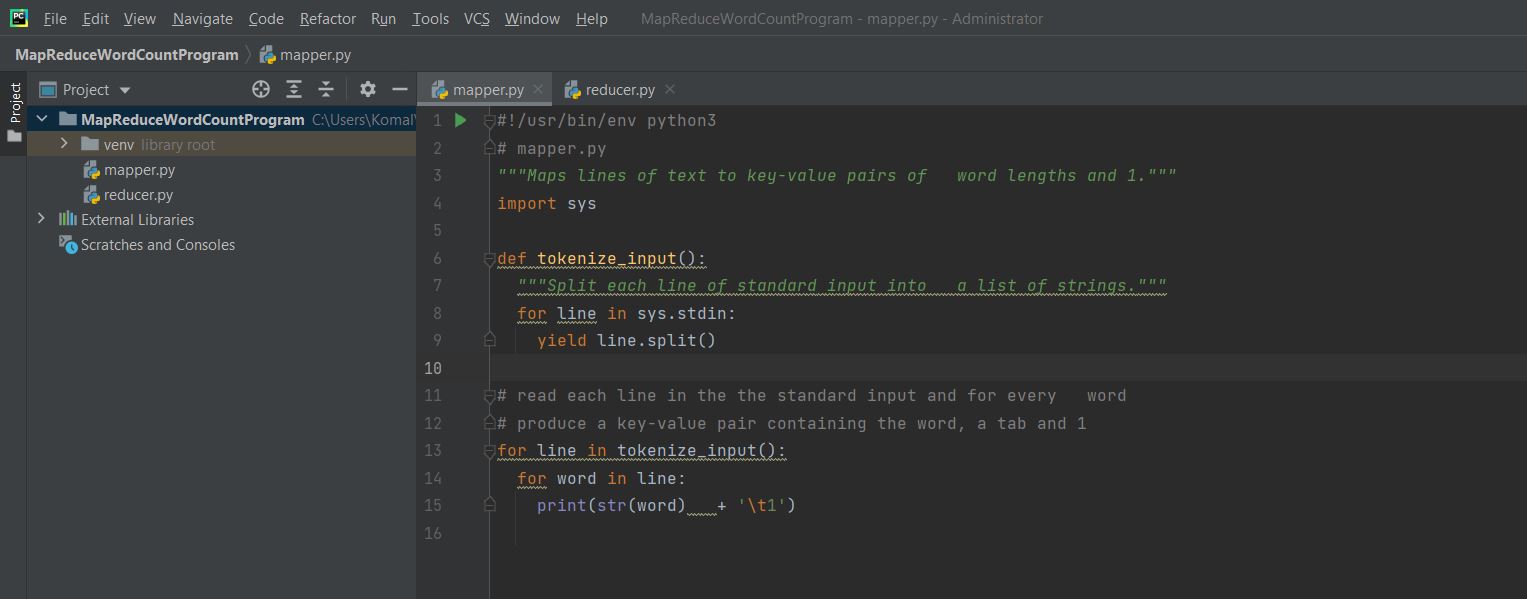
# Development of MapReduce Word Count Program in Python

## I developed the MapReduce Word Count Program in Python programming language PyCharm Community Edition software so install PyCharm if not installed from <https://www.jetbrains.com/pycharm/download>

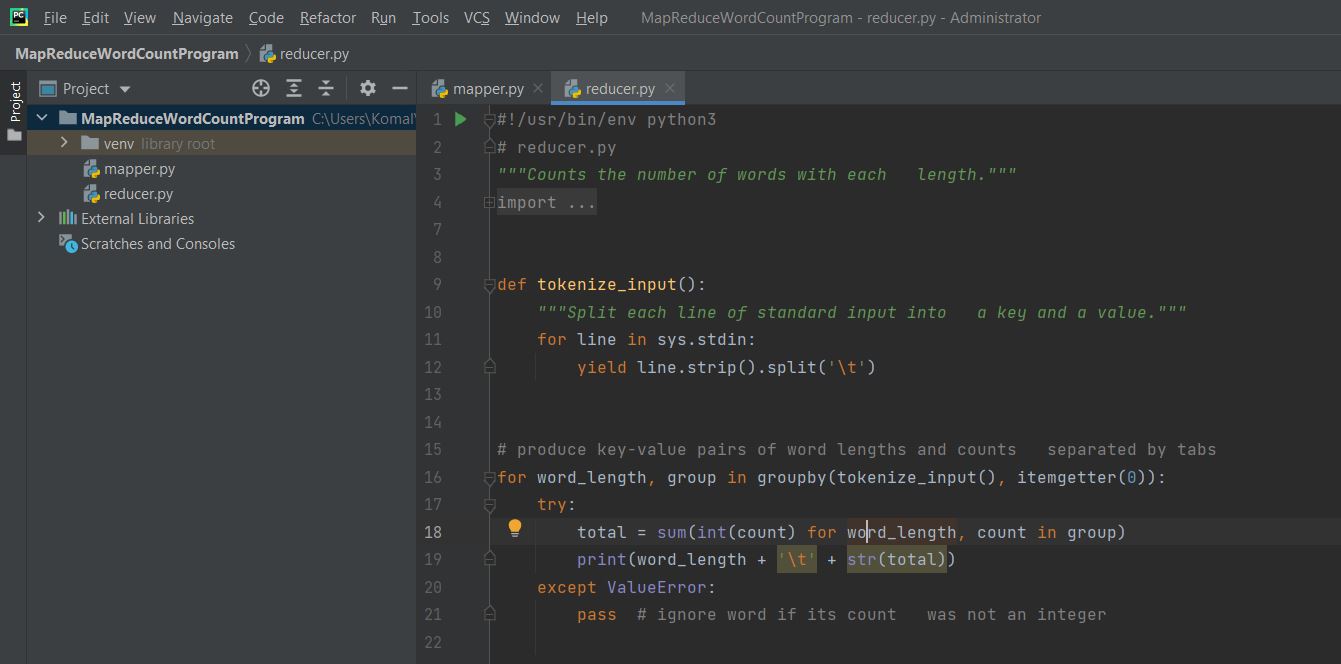
## Created a new project “MapReduceWordCountProgram” and developed mapper.py and reducer.py



1. Code of mapper.py

****

1. Code of reducer.py

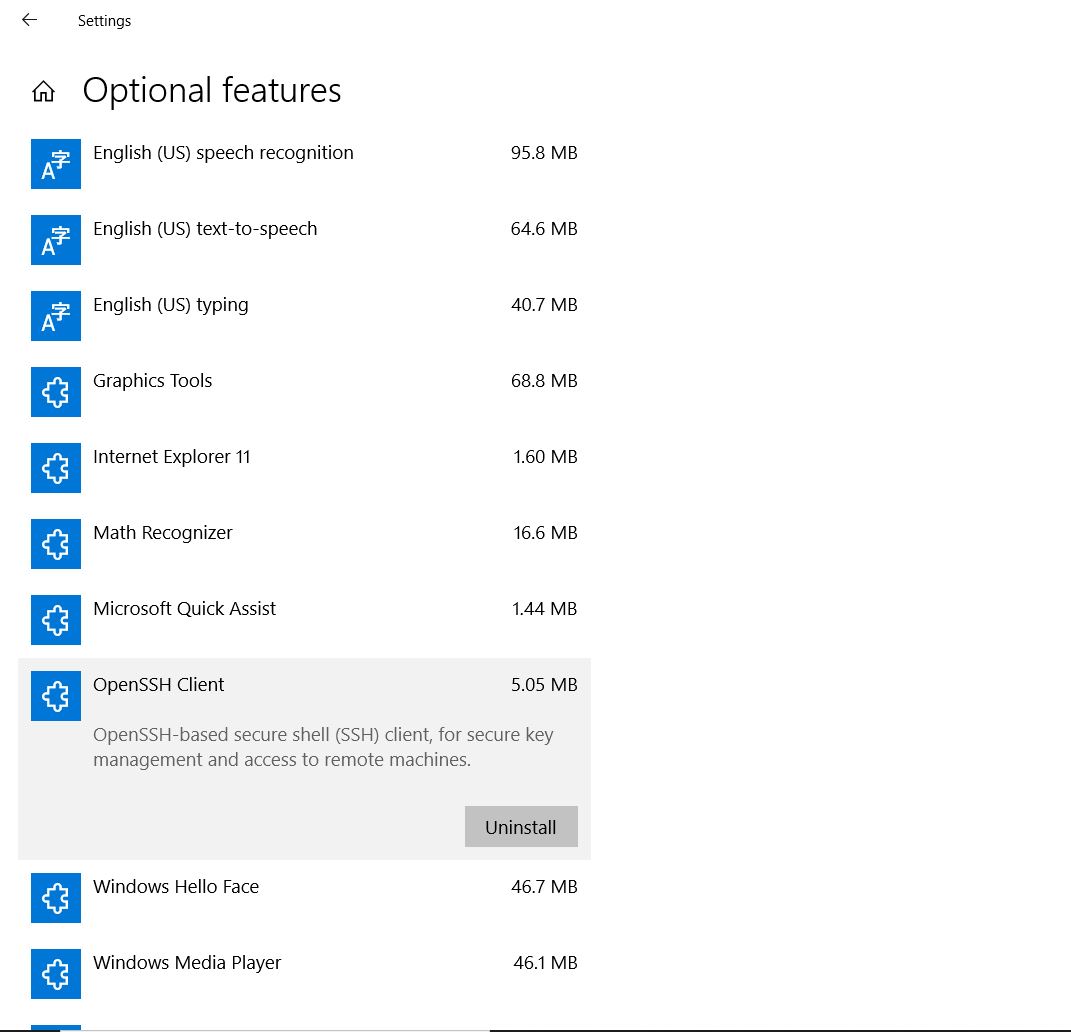


# Installation of OpenSSH Client & Server

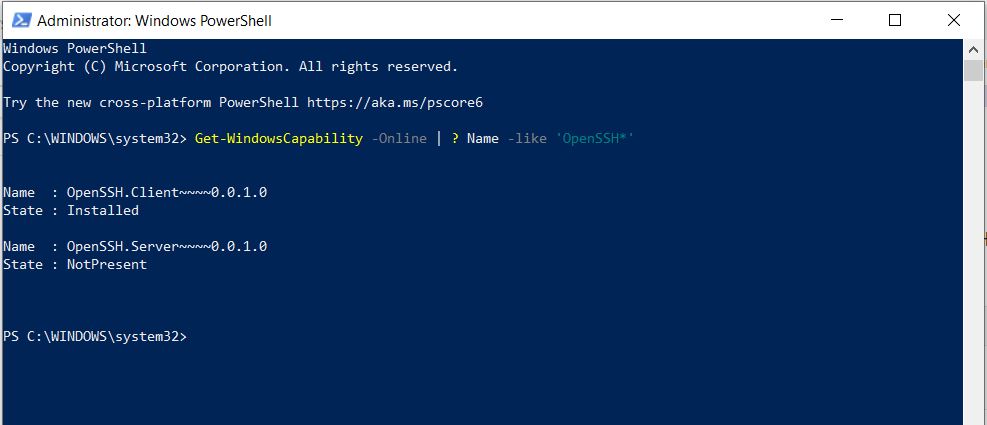
# To execute the MapReduce program there has to be OpenSSH Client & Server installed

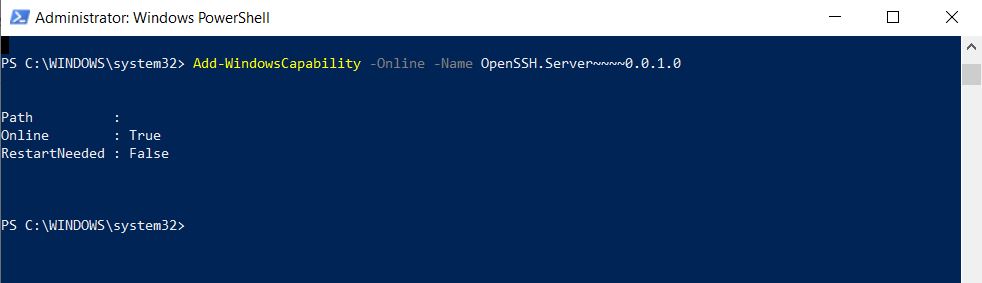
# Check whether OpenSSH client and server is present on Windows 10 or not using below steps

# Windows => Settings => Apps => Apps & Features => Optional features => OpenSSH

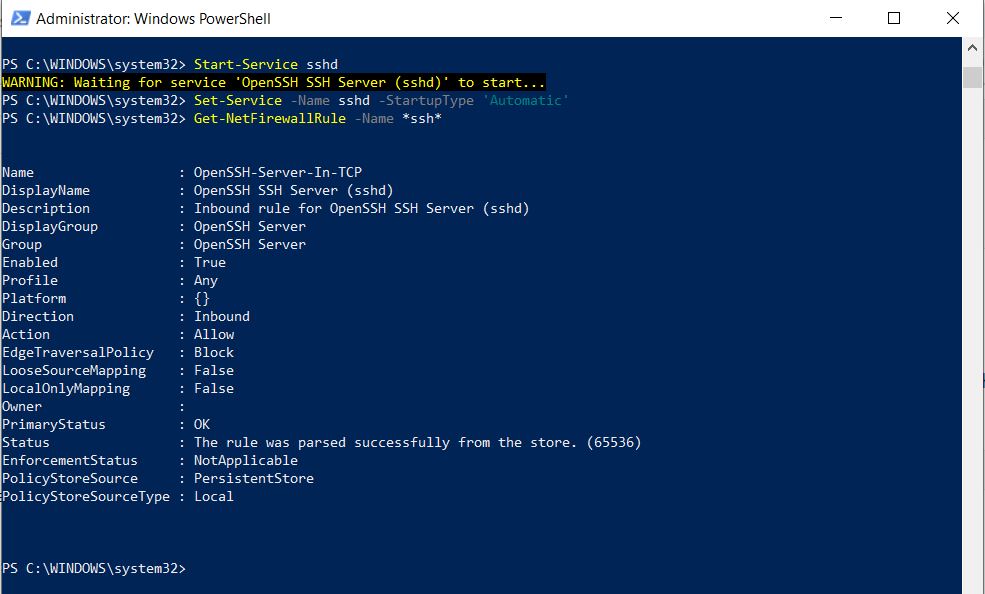
****

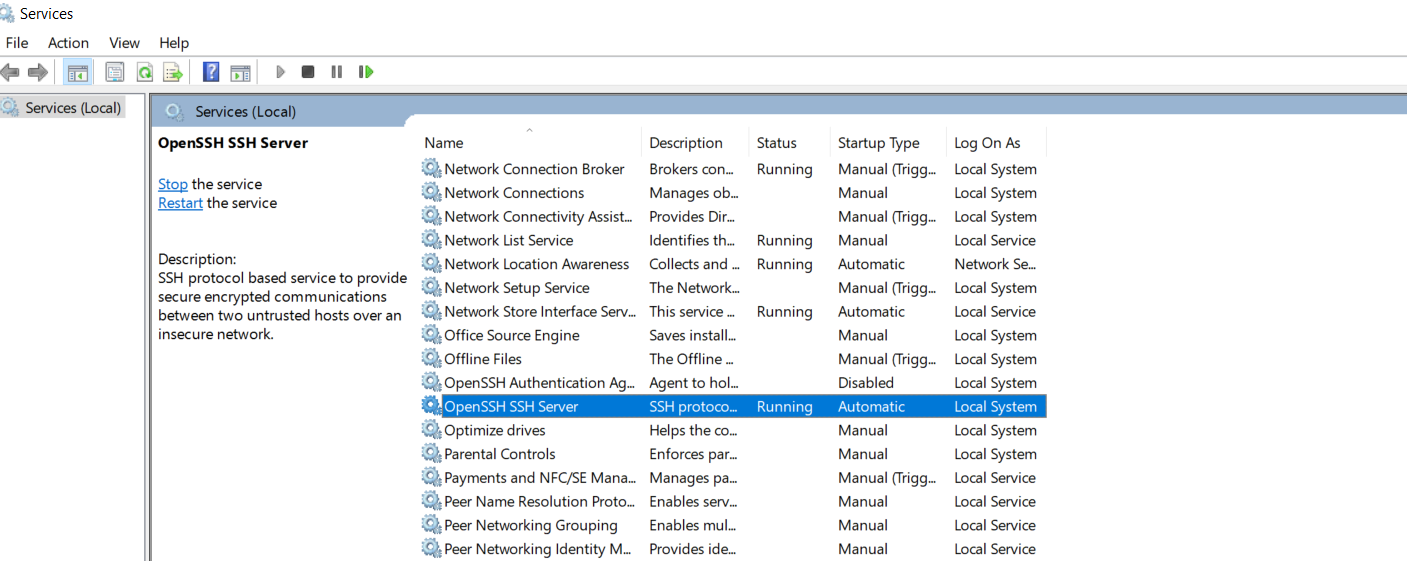
OpenSSH Client is already installed but OpenSSH Server is not installed so install it using Windows Powershell



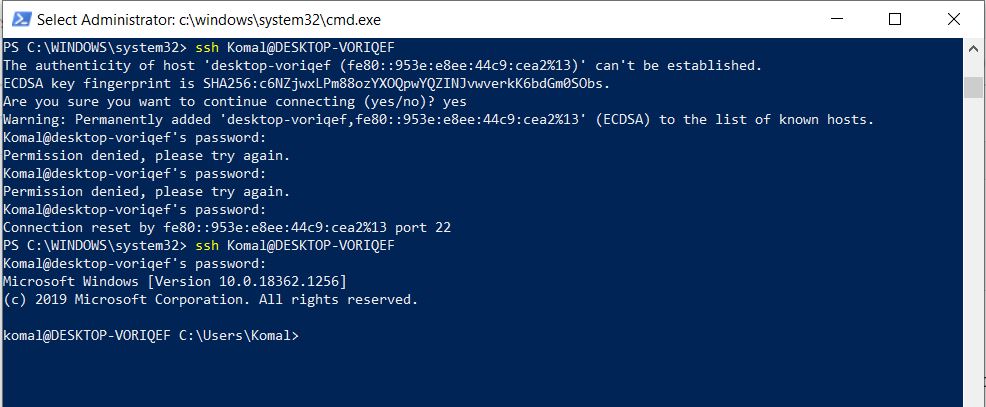


1. Start and Configure SSH and it’s services from services.msc



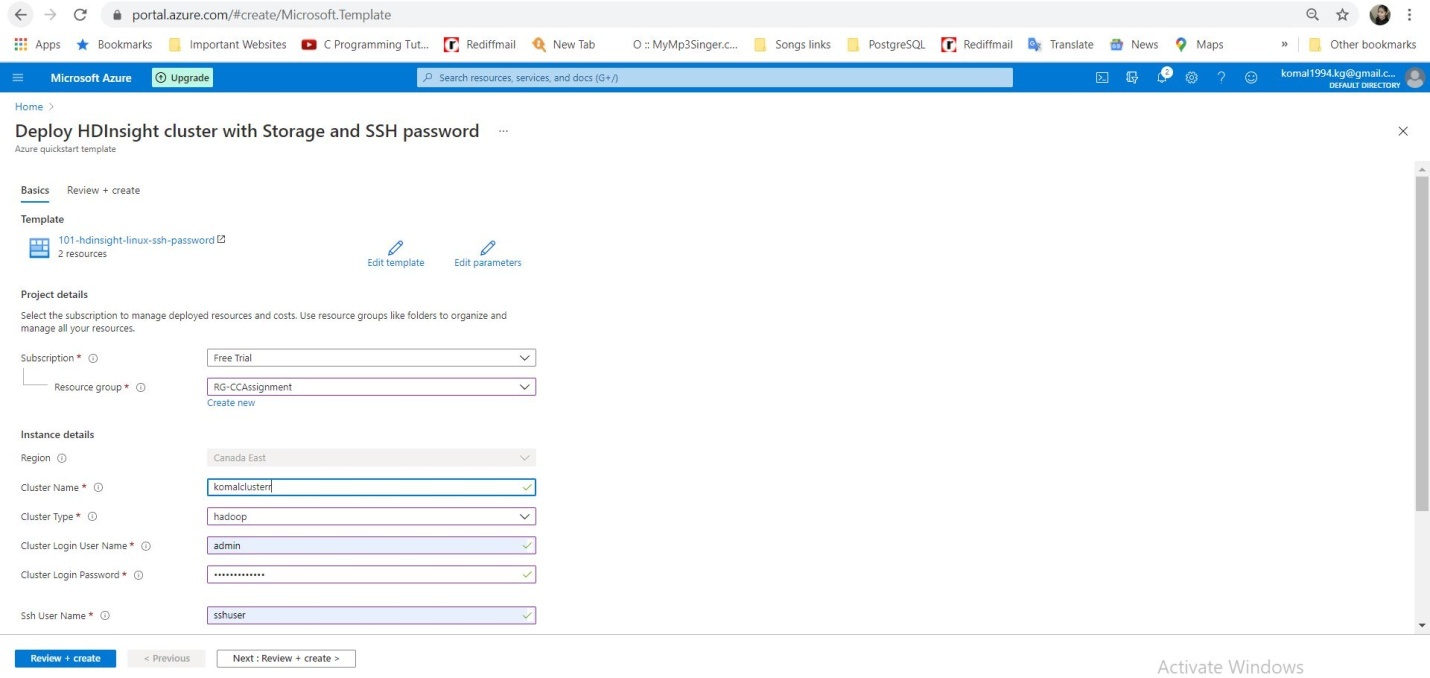


1. SSH is connected

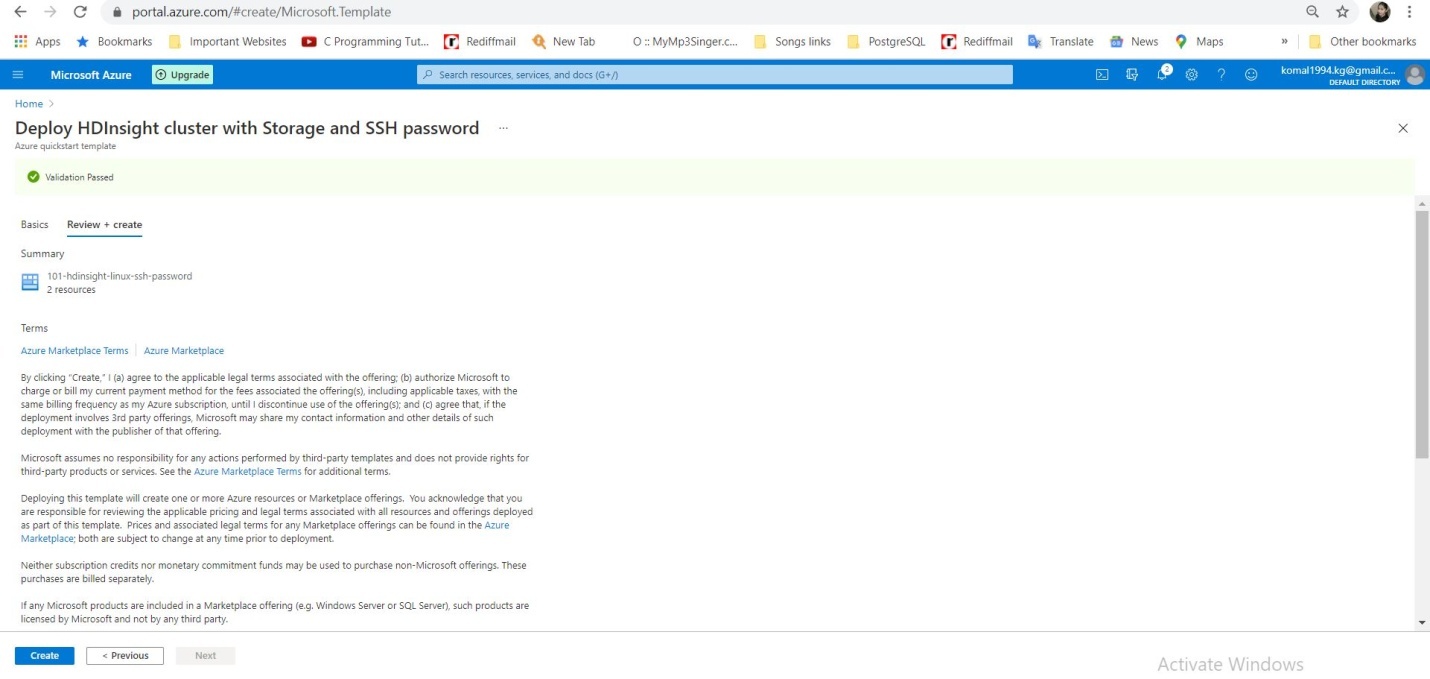


# Creation of Azure HDInsight Hadoop Cluster

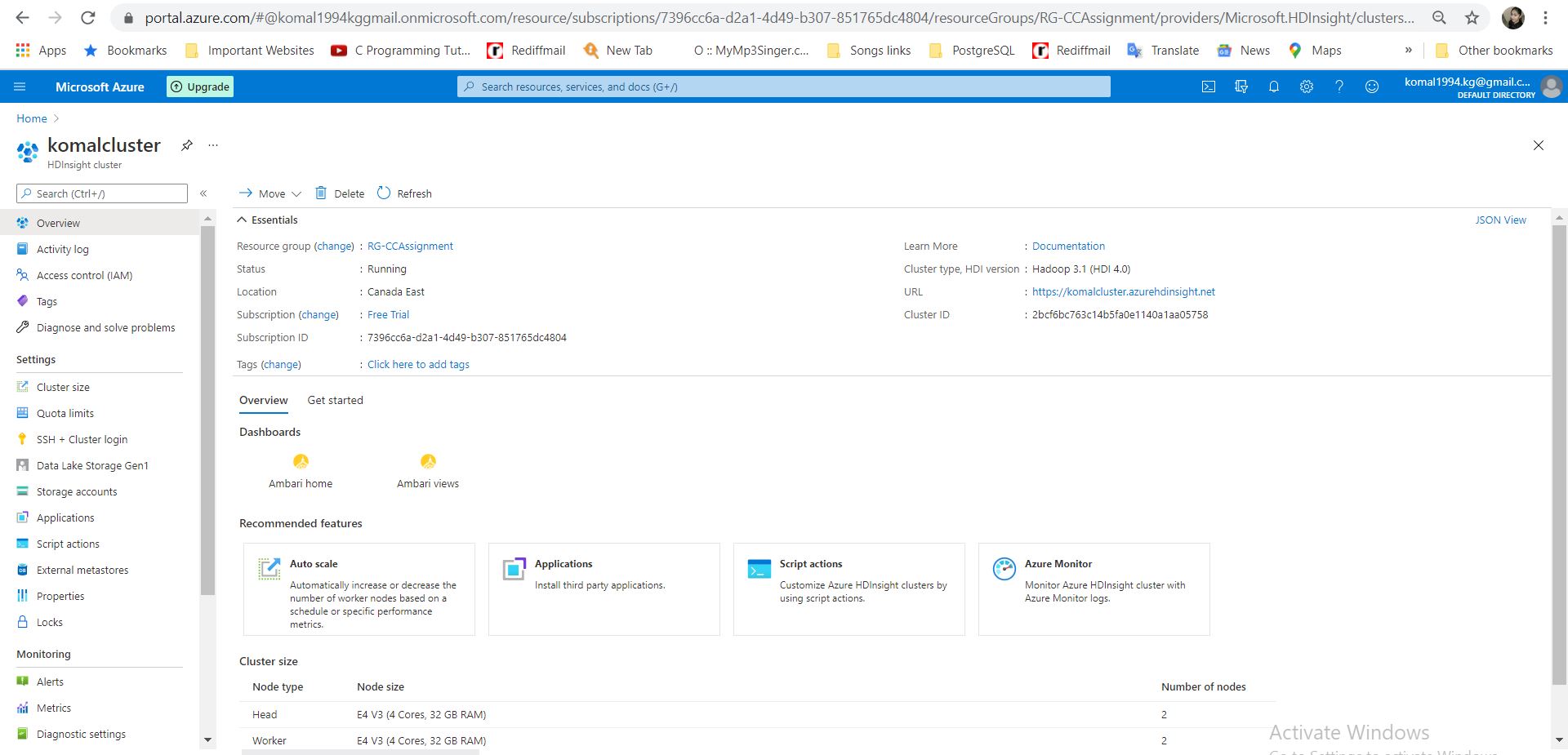
1. Create an Microsoft Azure account from [**https://azure.microsoft.com/en-in/free/cloud-services/search/**](https://azure.microsoft.com/en-in/free/cloud-services/search/)
2. Now sign-in to Azure account and create Azure HDInsight Hadoop Cluster named as **“komalcluster”** from [**https://portal.azure.com/#create/Microsoft.Template**](https://portal.azure.com/#create/Microsoft.Template) and click on **Review+Create**



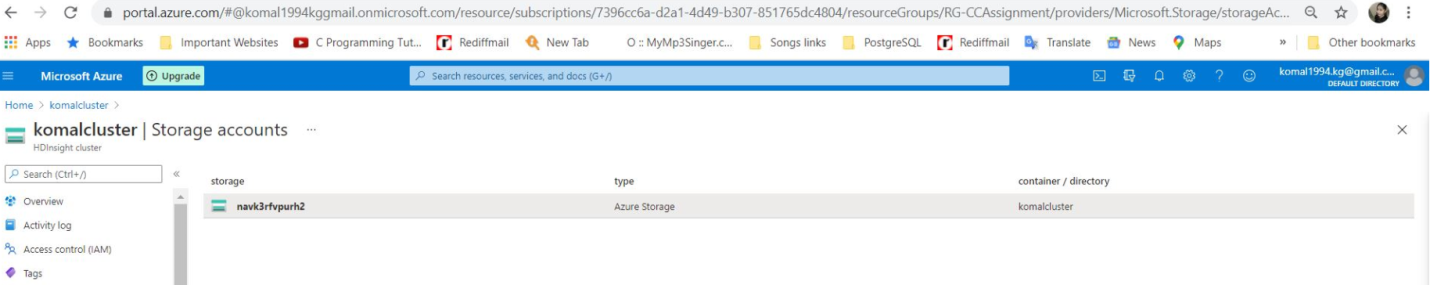
1. Click on **Create**



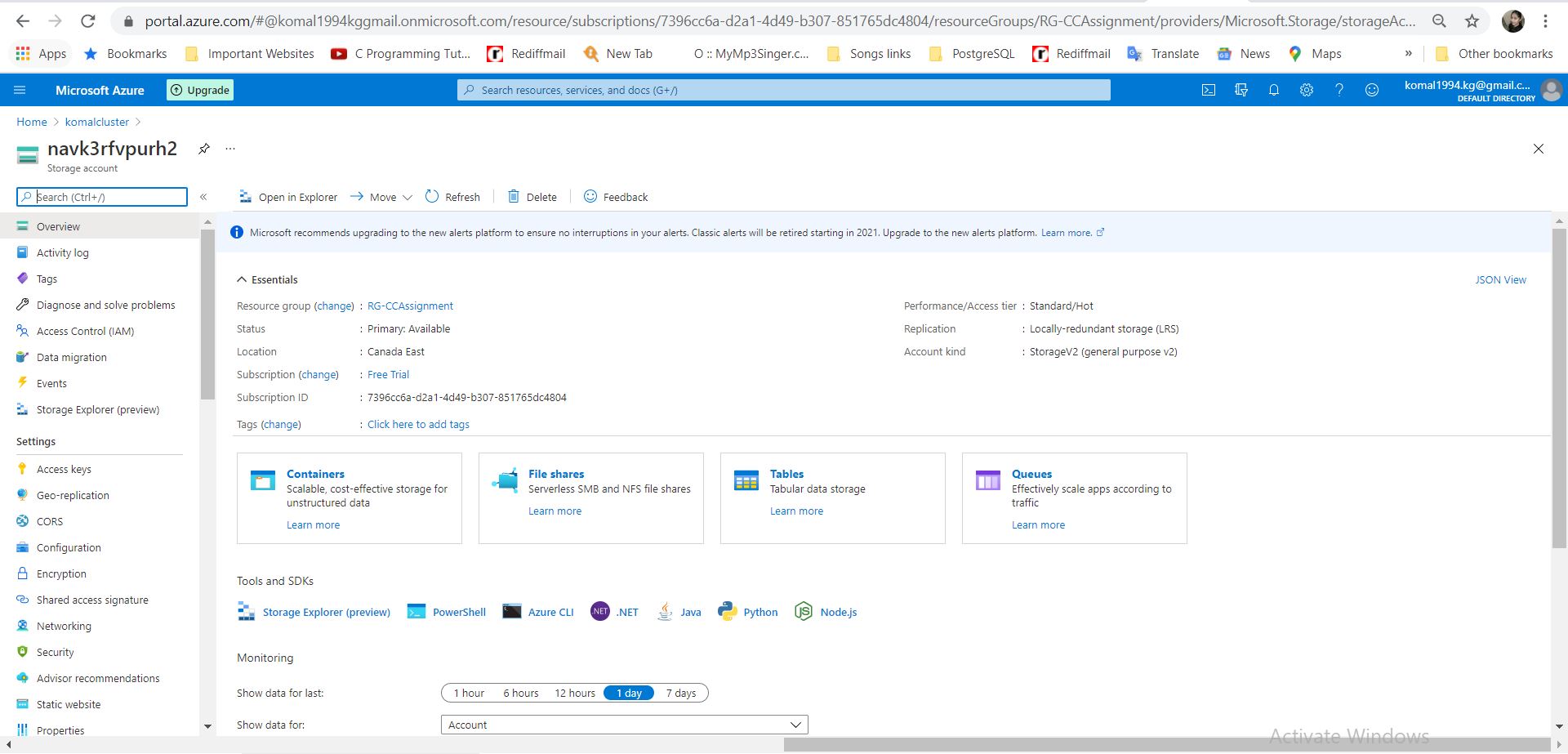
1. **“komalcluster”** Overview



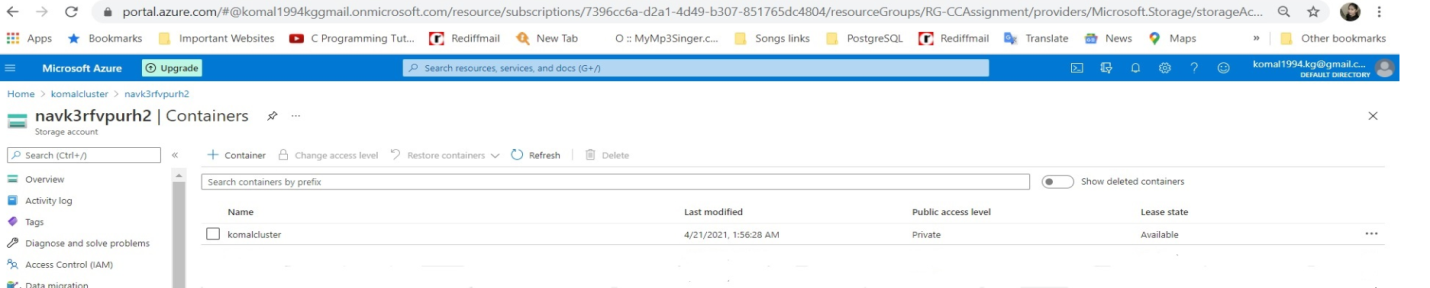
1. Go to komalcluster default “**Azure** **Storage Account”** and click on Storage Account as shown in below screenshot.



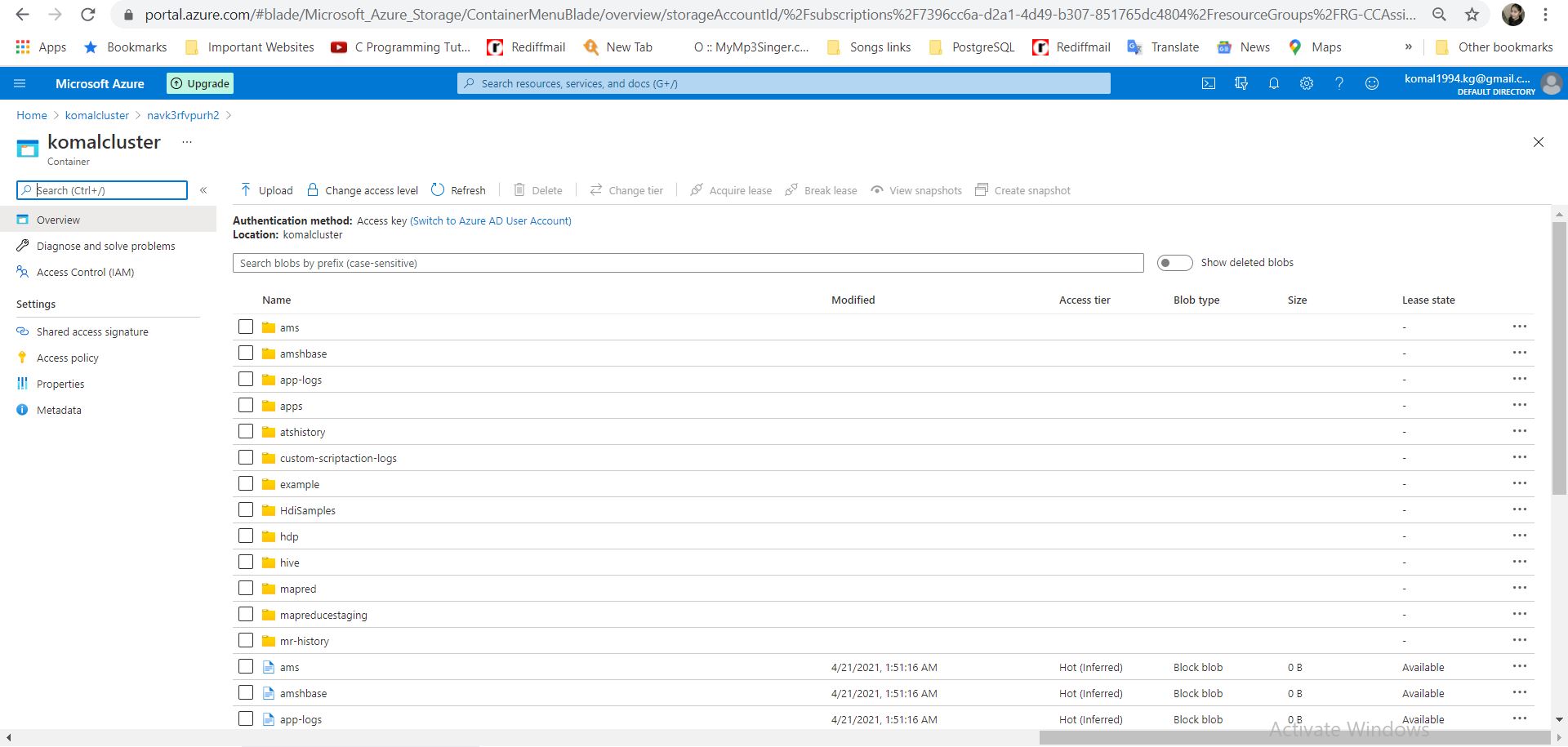
1. Click on **“Containers”**



1. Click on “**komalcluster”** default container

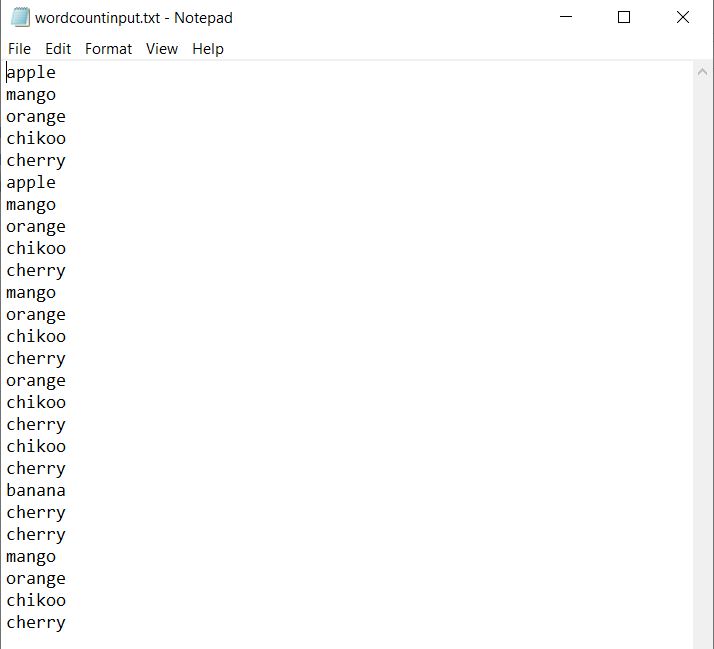


1. **“komalcluster”** overview and HDInsight Hadoop Cluster Configuration is done in Azure.



# Program Execution

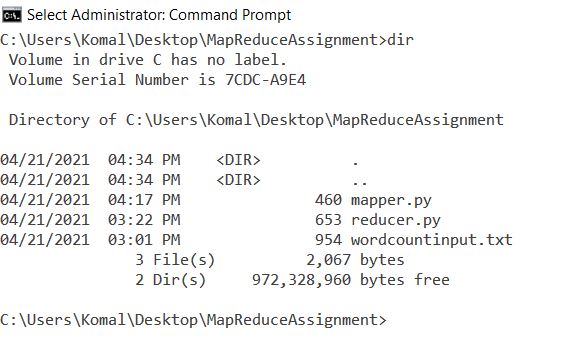
1. Create one input file wordcountinput.txt

****

1. Now move mapper.py, reducer.py and wordcountinput.txt into 1 local folder as below

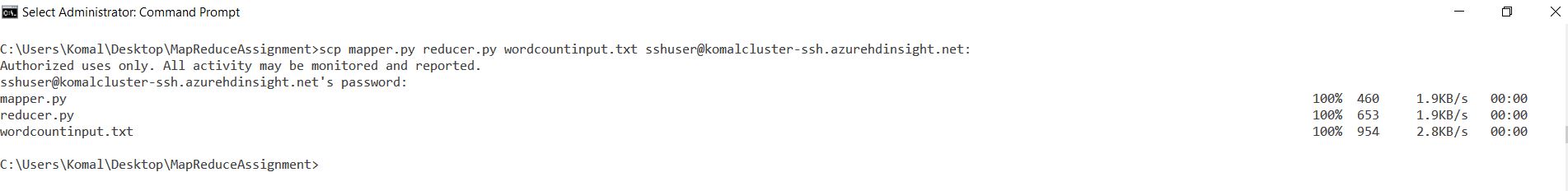
****

1. Now open Command Prompt or Windows Powershell to execute the MapReduce Word Count Program and go to the above directory where 3 files are copied.



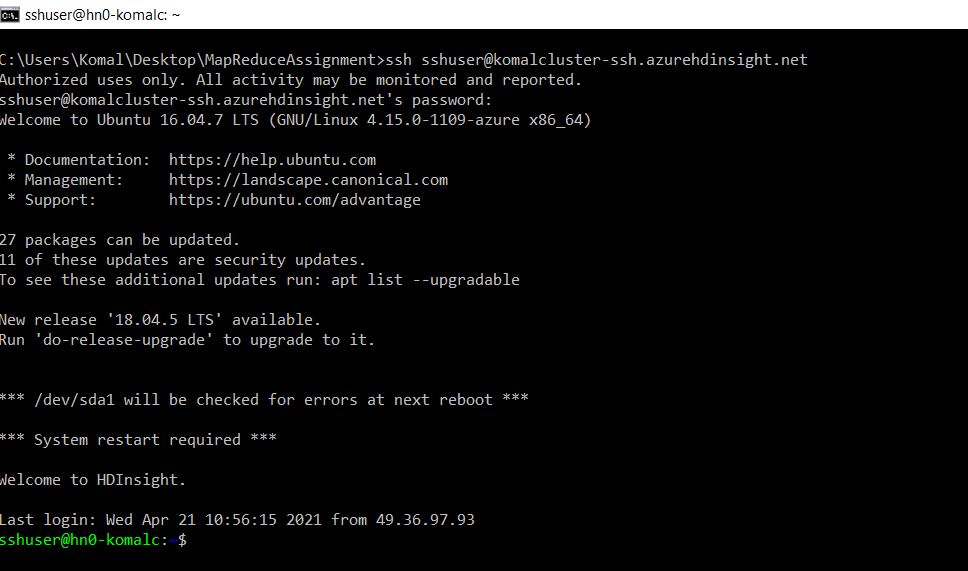
1. Now copy these 3 files to the HDInsight Hadoop cluster using command

**“scp mapper.py reducer.py wordcountinput.txt** [**sshuser@komalcluster-ssh.azurehdinsight.net**](mailto:sshuser@komalcluster-ssh.azurehdinsight.net)**:”**

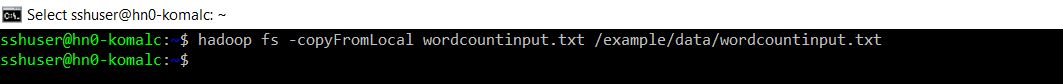


1. Now conncet HDInsight Hadoop Cluster using commad

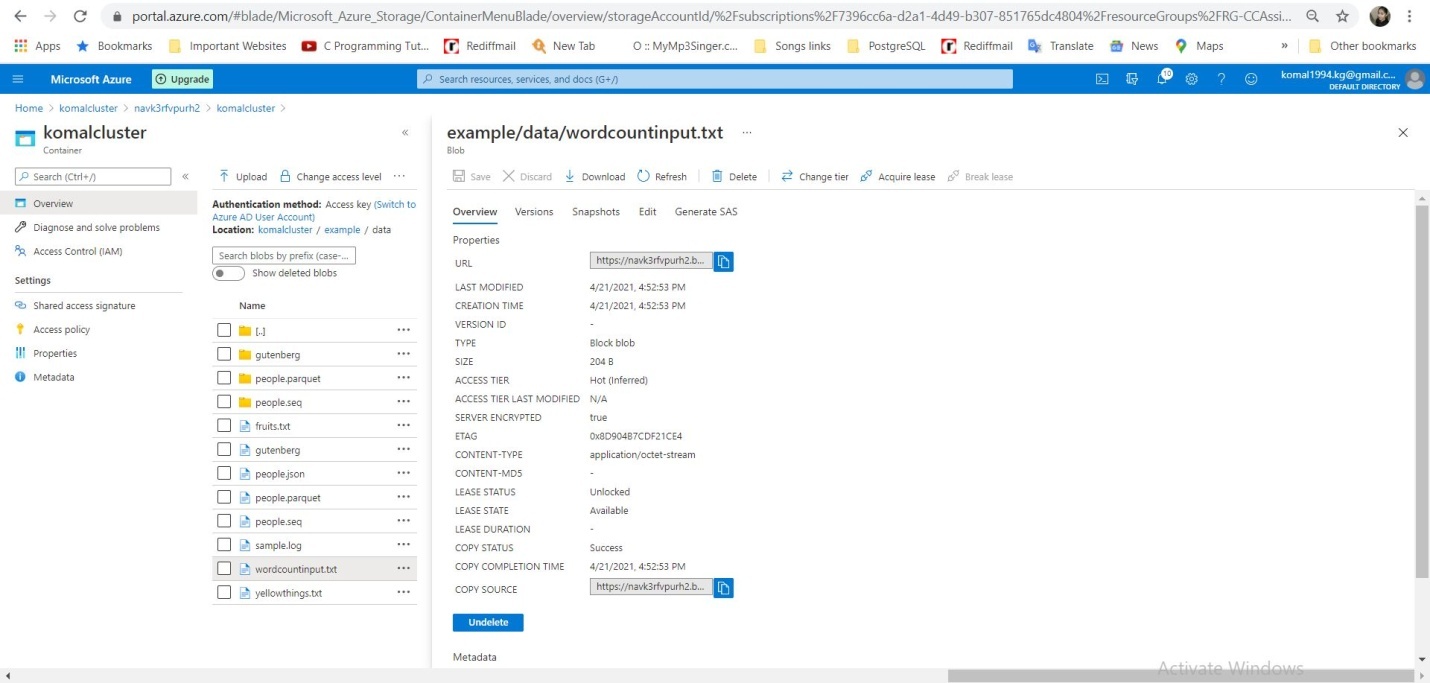
**“ssh** [**sshuser@komalcluster-ssh.azurehdinsight.net**](mailto:sshuser@komalcluster-ssh.azurehdinsight.net)**”**



1. Now copy wordcountinput.txt file to Hadoop using below command

**“hadoop fs -copyFromLocal wordcountinput.txt /example/data/wordcountinput.txt”**

1. Check into Azure portal that whether wordcountinput.txt file has copied in **komalcluster** **container** of HDInsight Hadoop Cluster **komalcluster**.



1. Now to execute the MapReduce job for developed python WordCount Program run below command

**yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar \**

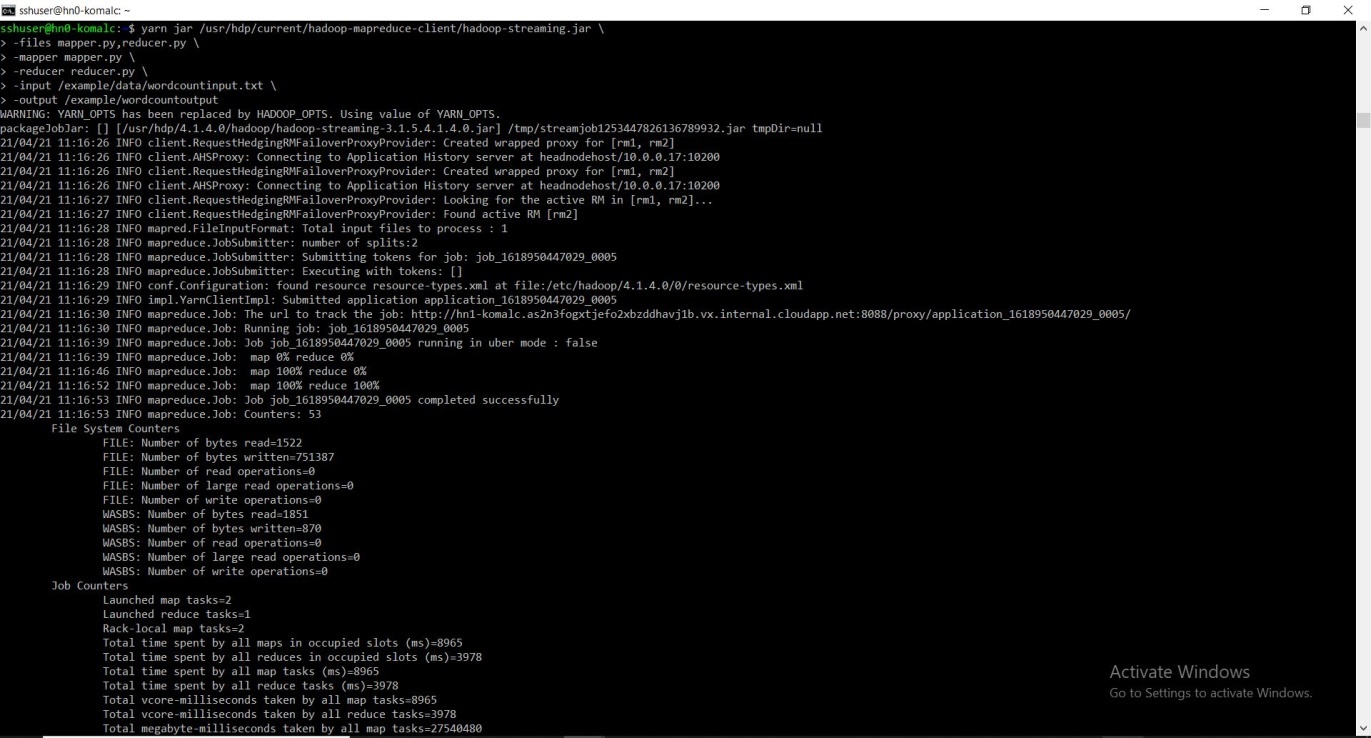
**-files mapper.py,reducer.py \**

**-mapper mapper.py \**

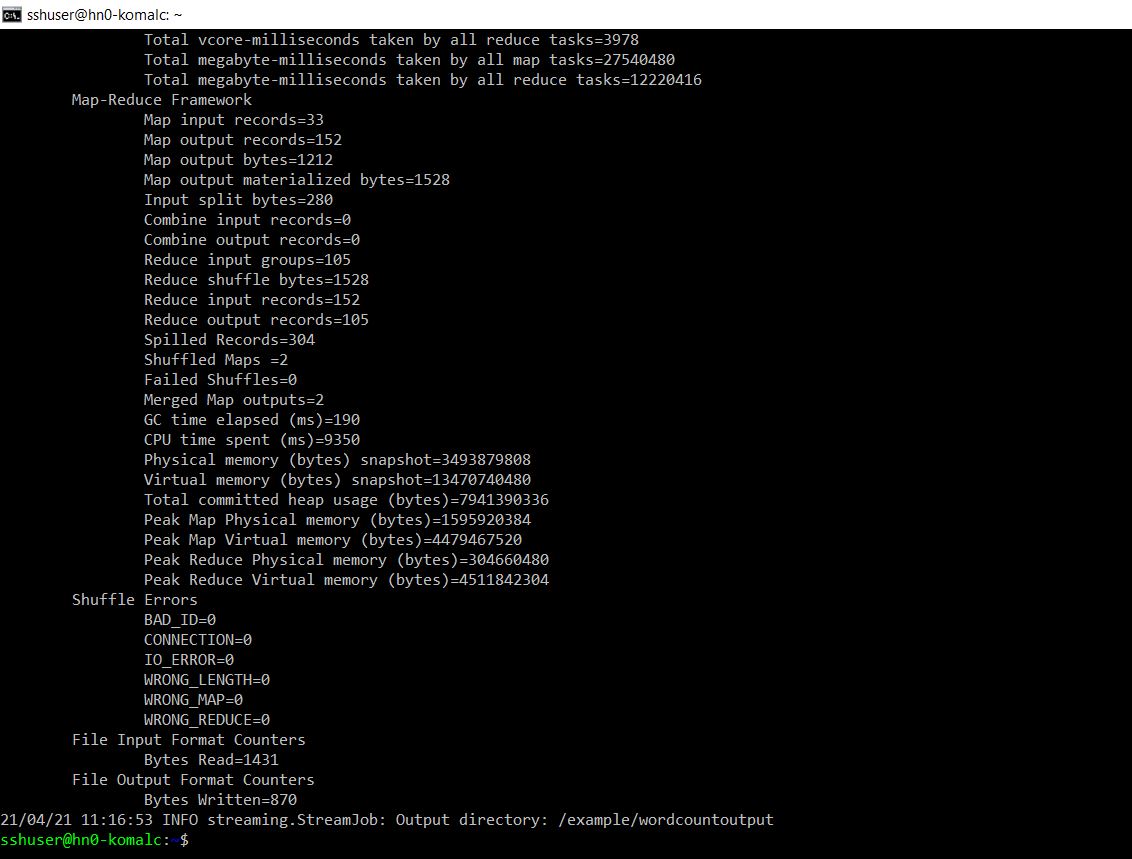
**-reducer reducer.py \**

**-input /example/data/wordcountinput.txt \**

**-output /example/wordcountoutput**

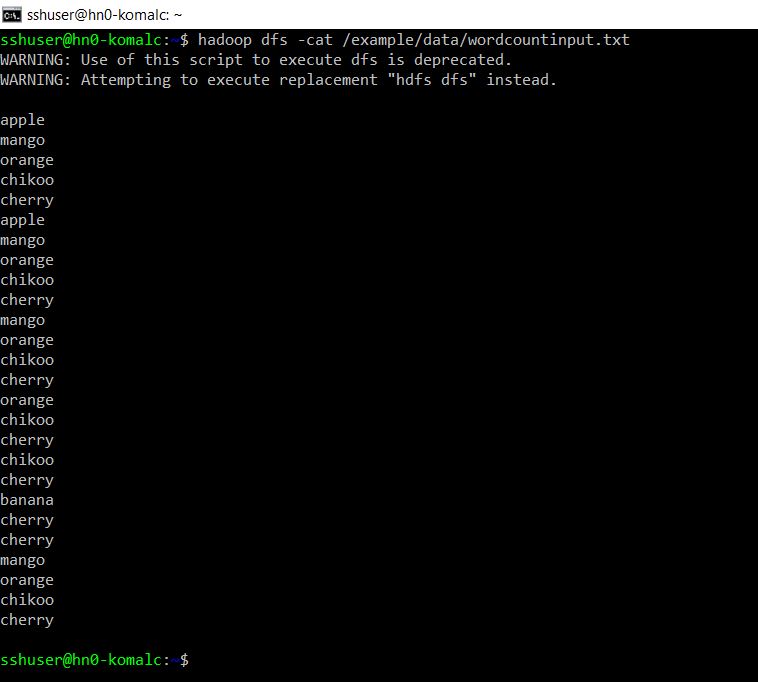
****

1. Job successfully executed

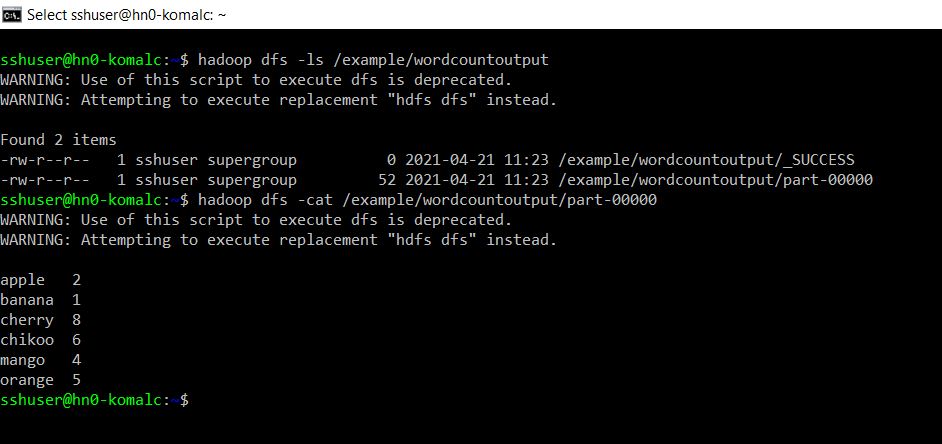
****

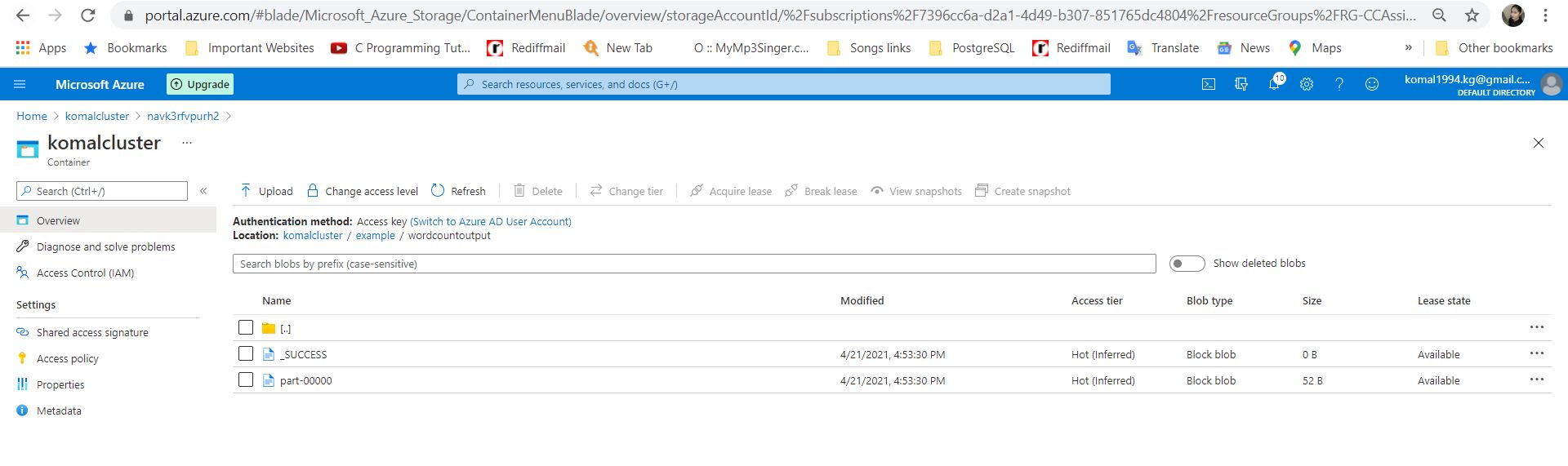
# Program Output

1. Input File



1. Output file in Command Prompt as well as into folder example/wordcountoutput of Azure Portal .



****

**Second Way: By installing Hadoop & Java on Windows 10 and then using in-build hadoop-mapreduce-examples.jar**

# Prerequisite softwares to be installed in Windows 10

## Java 8

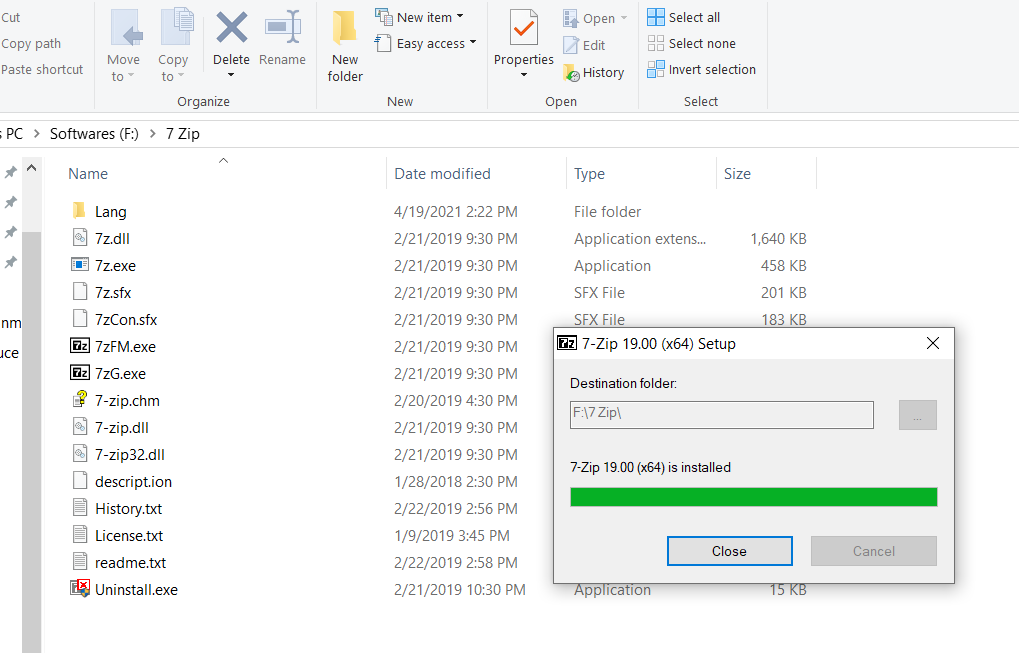
## Hadoop 3.2.2

## To unzip downloaded Hadoop binaries, we should install [7zip](https://www.7-zip.org/download.html).

## Administrator access to run the commands.

# Installation of 7 Zip

1. Download the 7 Zip 19.0 from <https://www.7-zip.org/> and install it

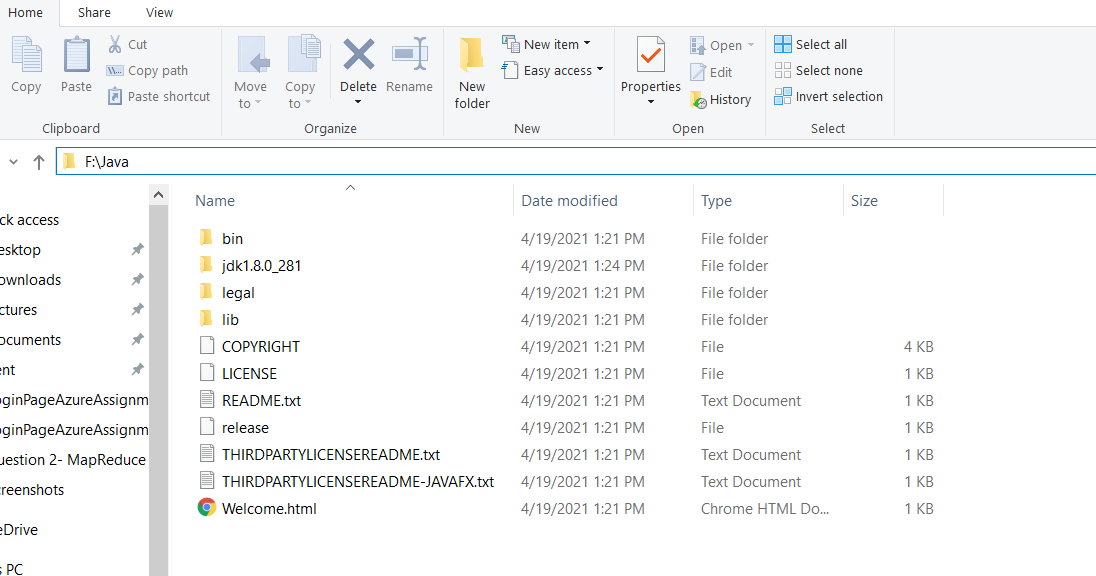


# Installation and Configuration of JAVA

## Installation

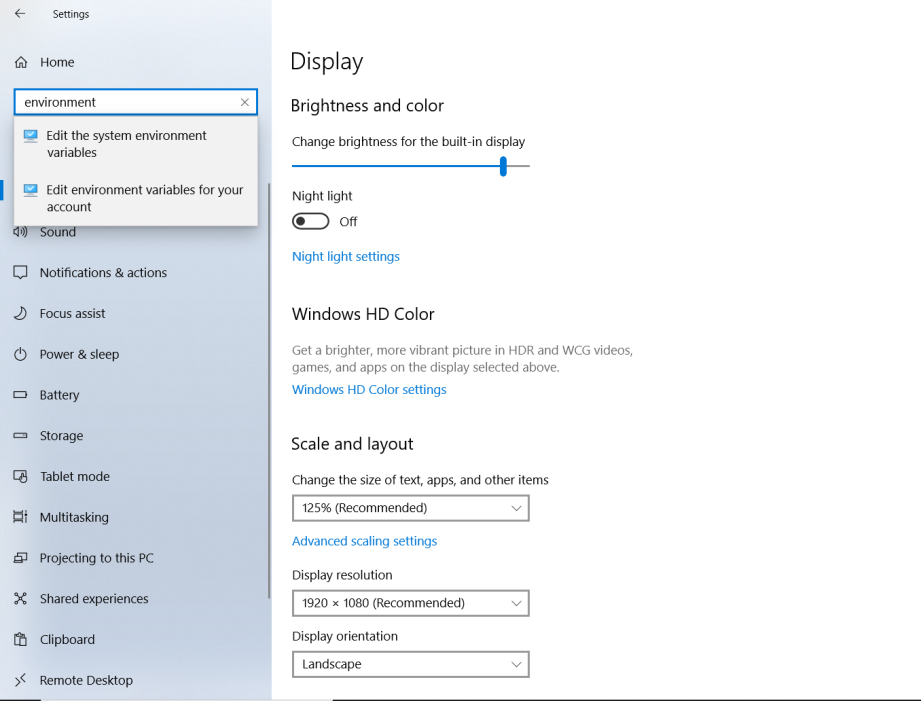
## Download java 8 from <https://www.oracle.com/in/java/technologies/javase/javase-jdk8-downloads.html> and install it. I have installed in F:\Java

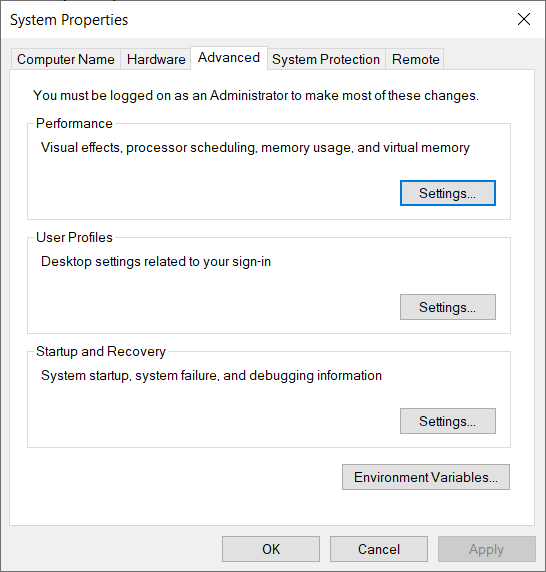
1. Movethe installed jdk1.8.0\_281 from C:\Program Files\Java to the folder F:\Java.



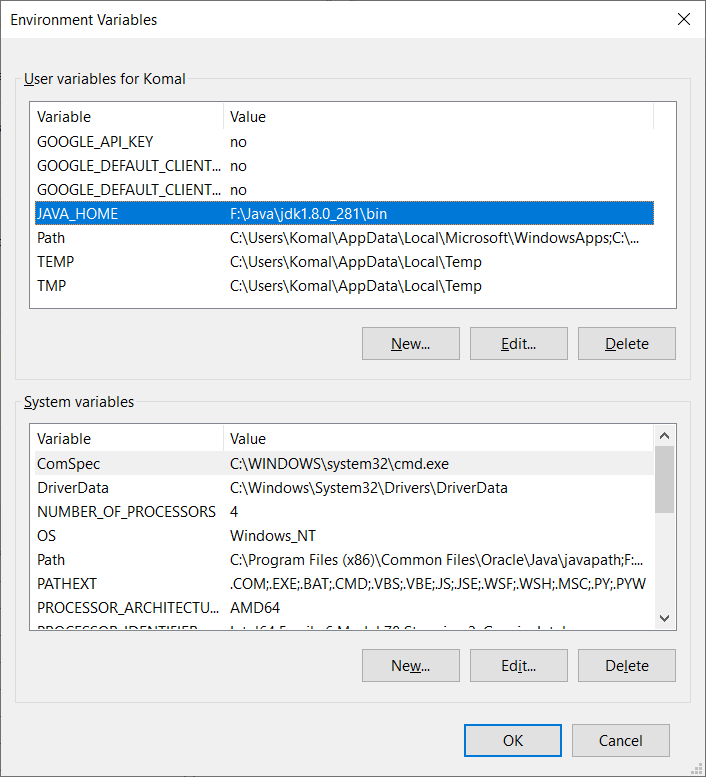
## Windows Configuration

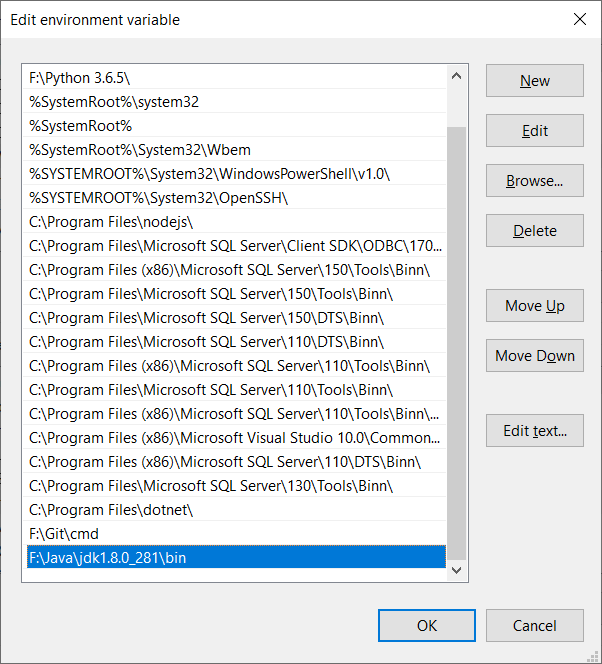
1. Go to Windows => Settings => System and type environment in search and select “Edit the system environment variables” as shown in screenshot below. Now select “Environment Variables”.



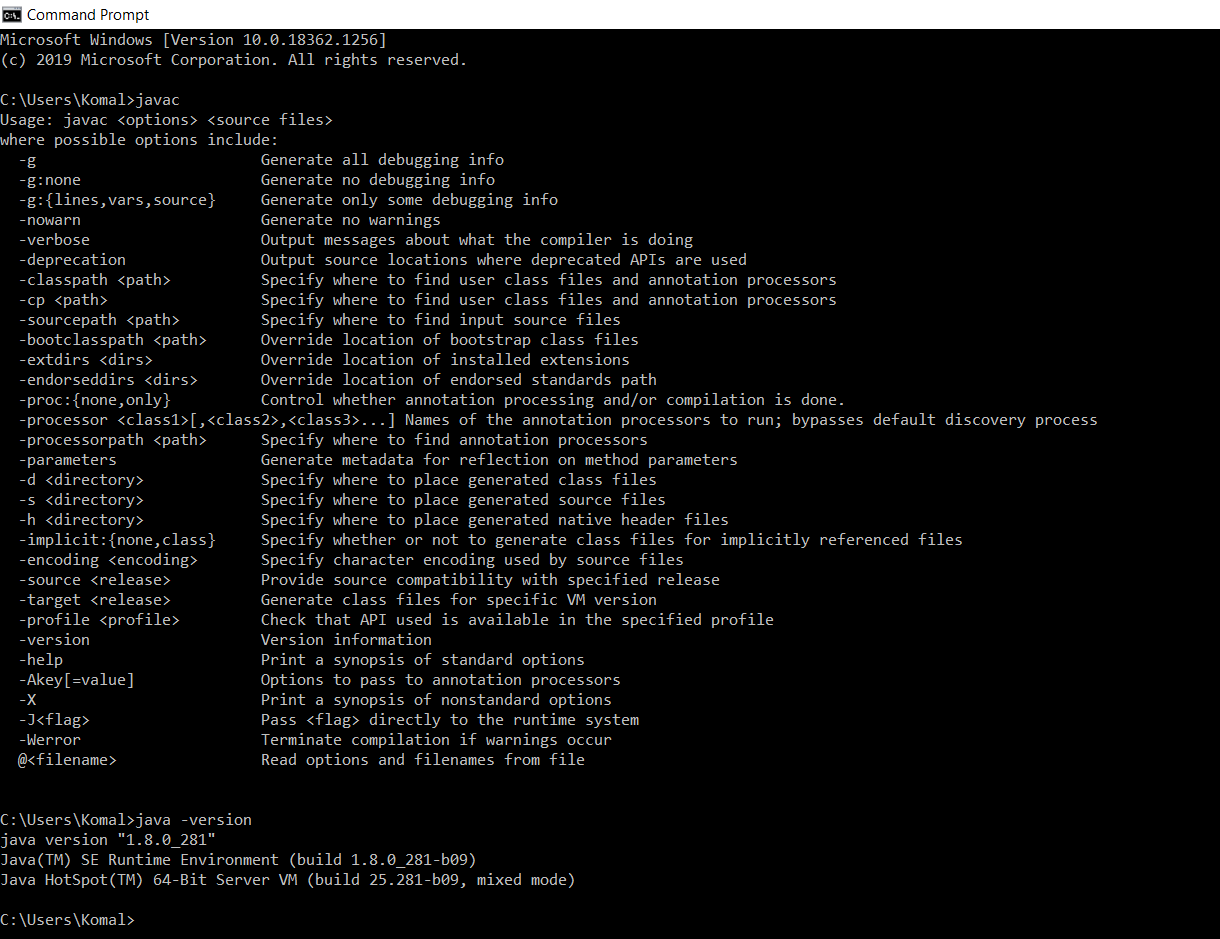


1. Now set the environment variable “JAVA\_HOME” as well as path for JAVA. After setting click on Ok.





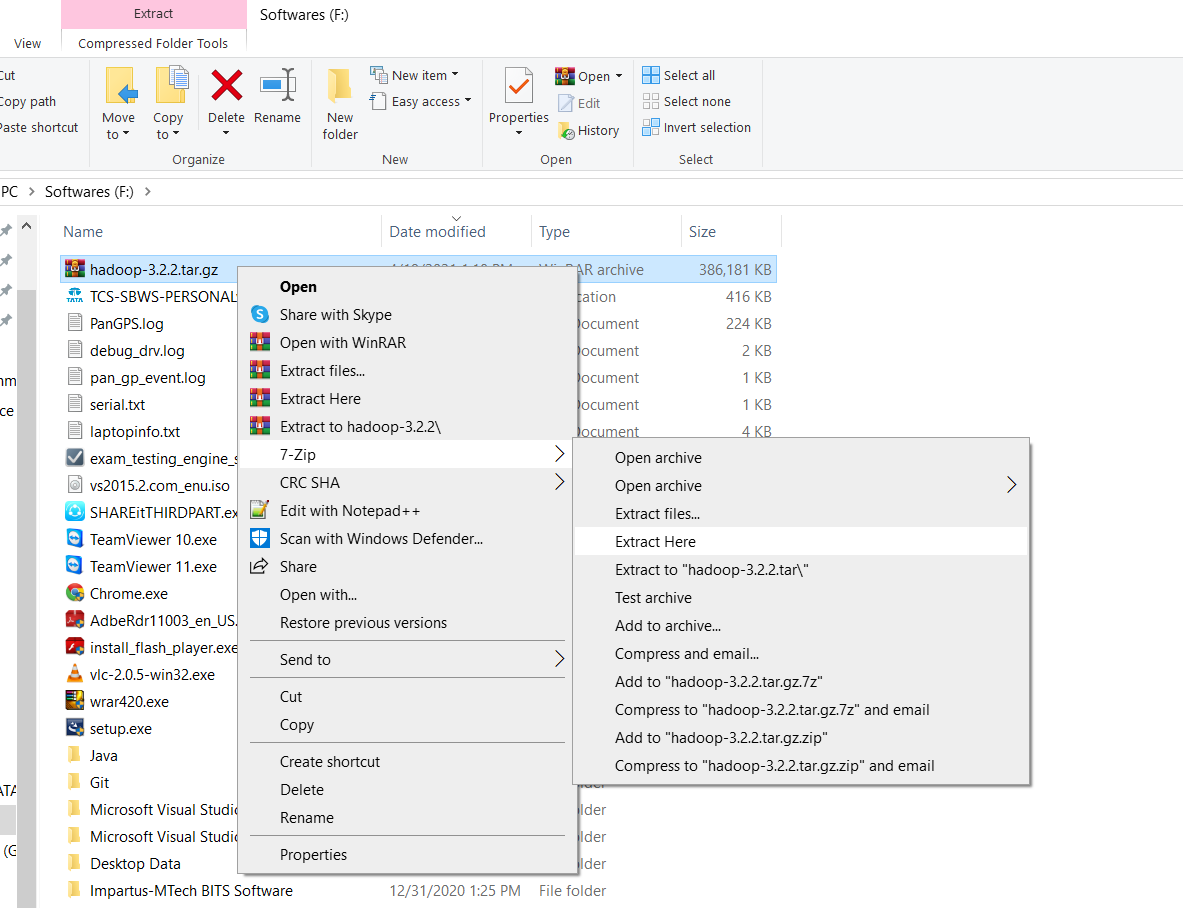
1. Now Java is successfully installed on our local system and to confirm it run the command **“javac”** and to check version run “**Java –version”** in command prompt.

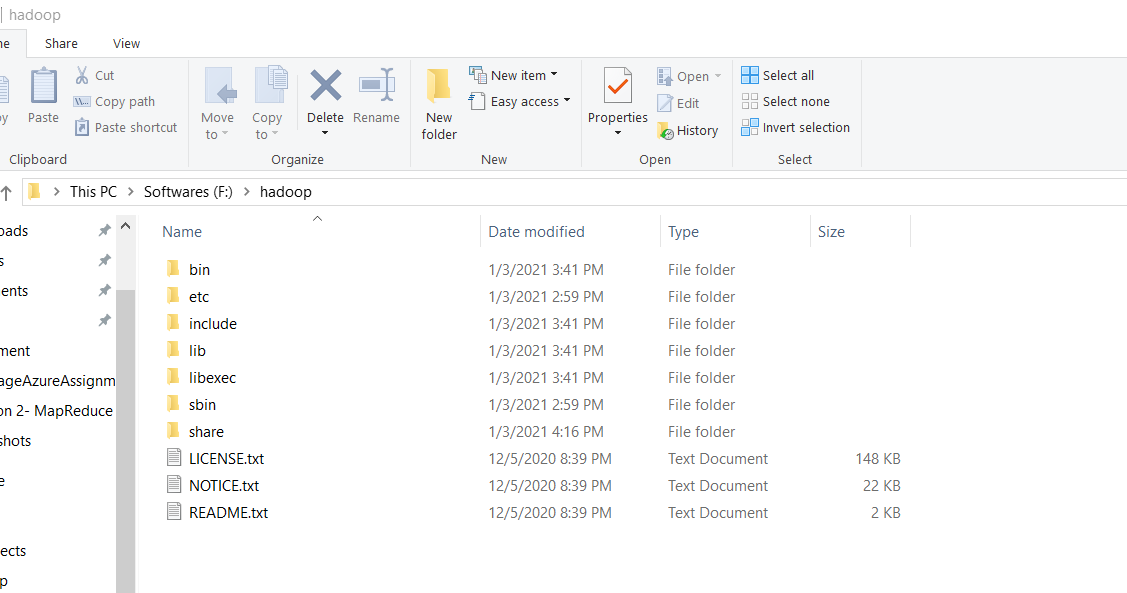


# Installation and Configuration of Hadoop

## Installation

1. Download Hadoop version 3.2.2 which is earlier to the latest version 3.3.0 from <https://hadoop.apache.org/releases.html> and select Binary link. Once selected Binary link , select mirror link to download hadoop tar file
2. Now unzip it using 7zip. I have unzip “**hadoop-3.2.2.tar.gz**” in F:\ and again unzip the **“hadoop-3.2.2.tar”**. Now rename unzipped tar folder to **“hadoop”.**



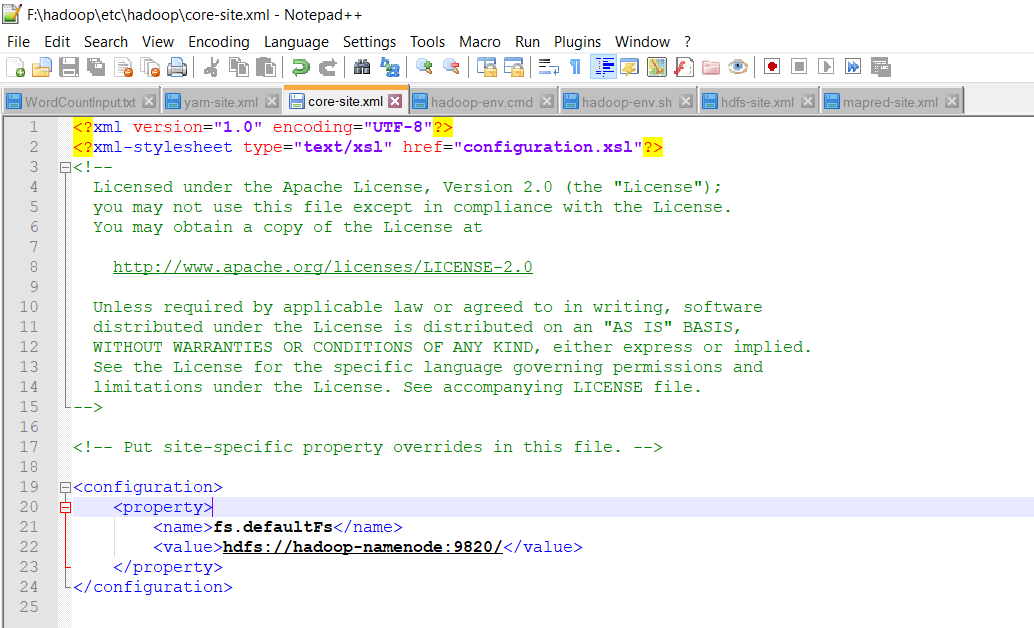


## Hadoop Configuration

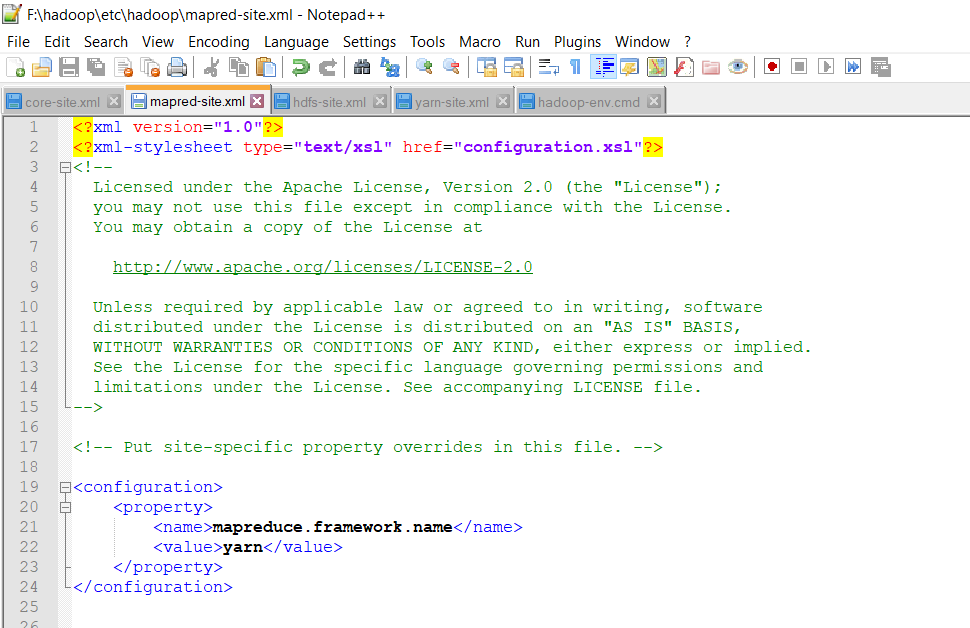
1. Now configure the 4 xml files inside hadoop/etc/hadoop folder

**core-site.xml , hdfs-site.xml, mapred-site.xml, yarn-site.xml**

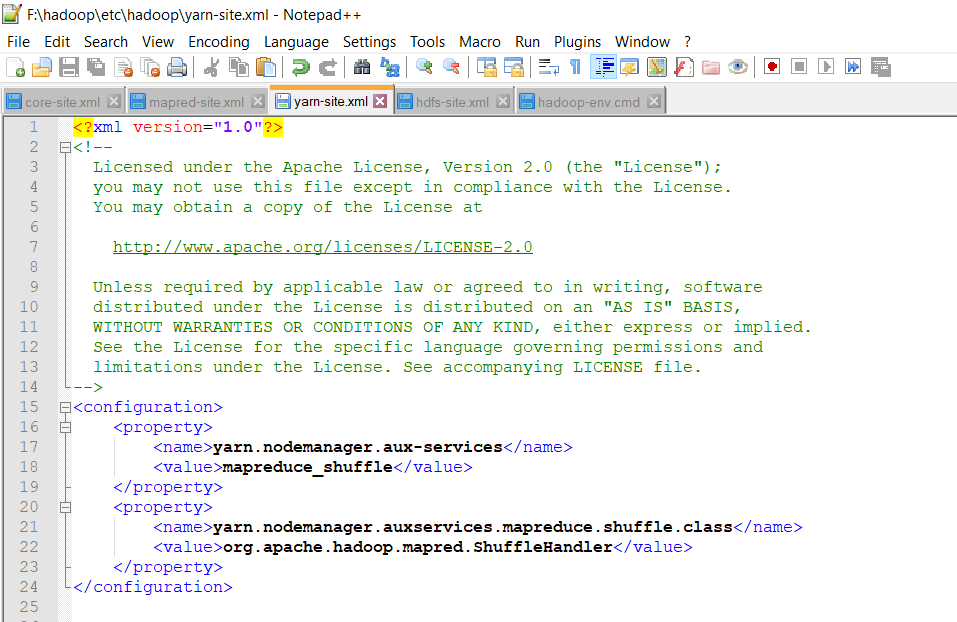
1. Configure core-site.xml and save it as below:



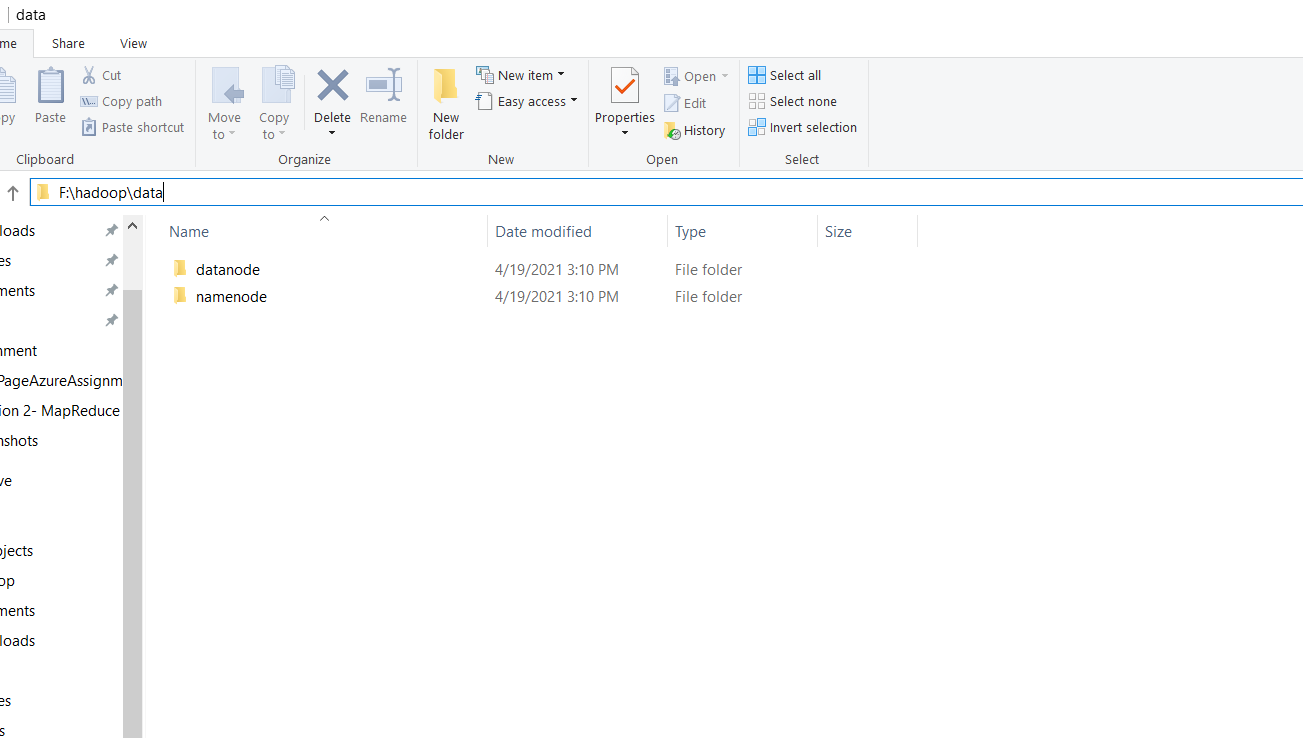
1. Configure mapred-site.xml and save it as below:



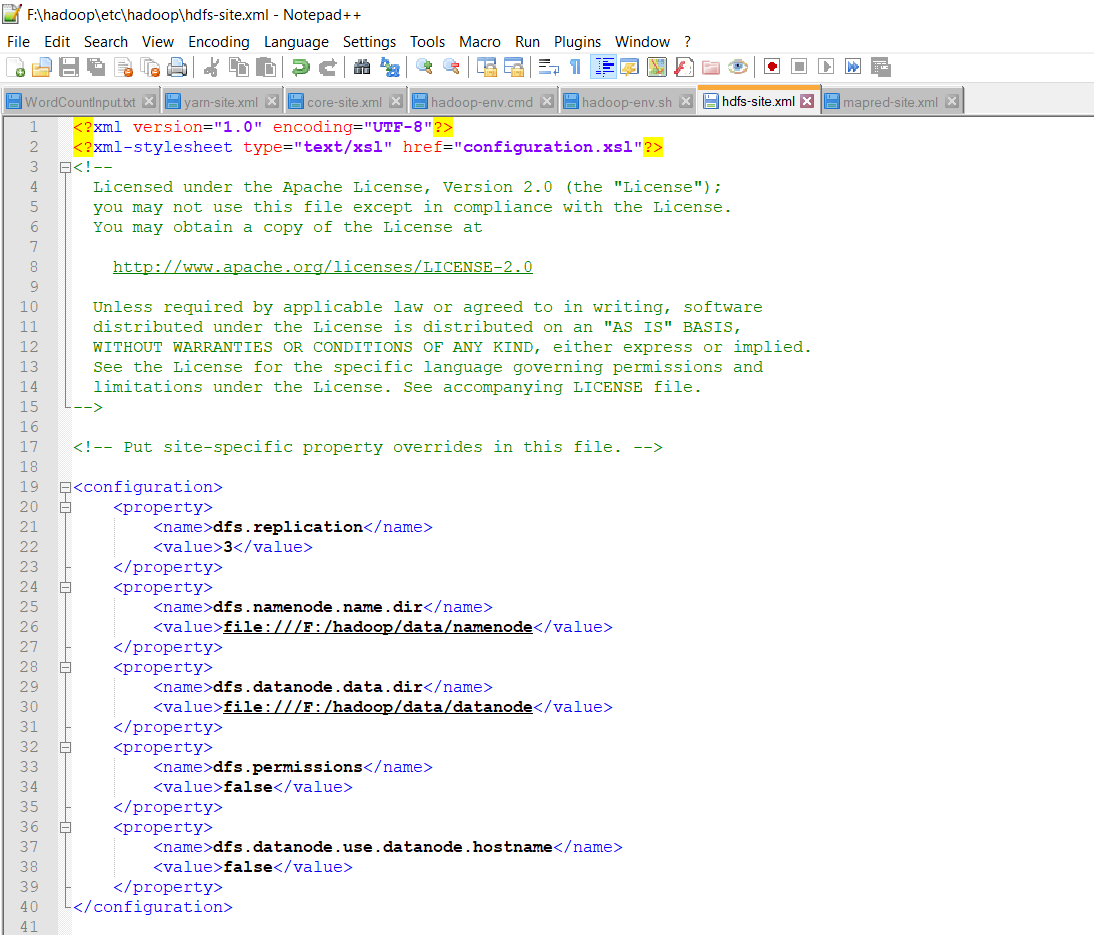
1. Configure yarn-site.xml and save it below:



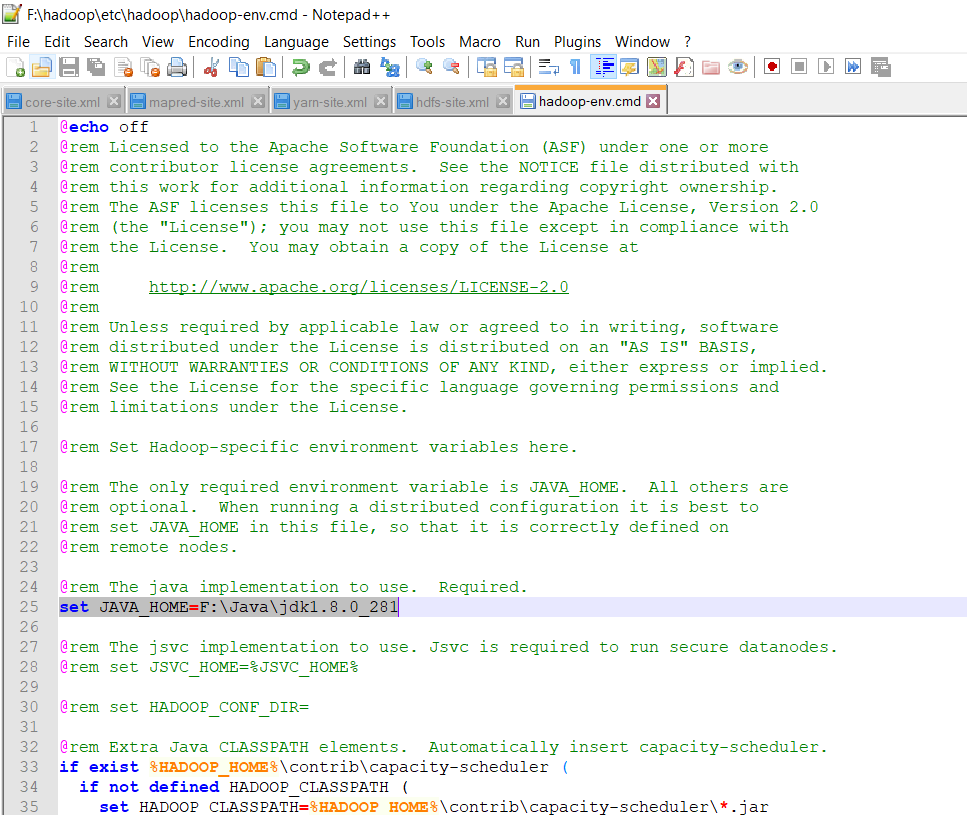
1. Before configuring hdfs-site.xml create one “**data”** folder inside **“F:\hadoop”** and inside this **“data”** folder create 2 more folders as **“namenode” and “datanode”** as below.



Now copy the path of folders namenode and datanode to configure the file hdfs-site.xml

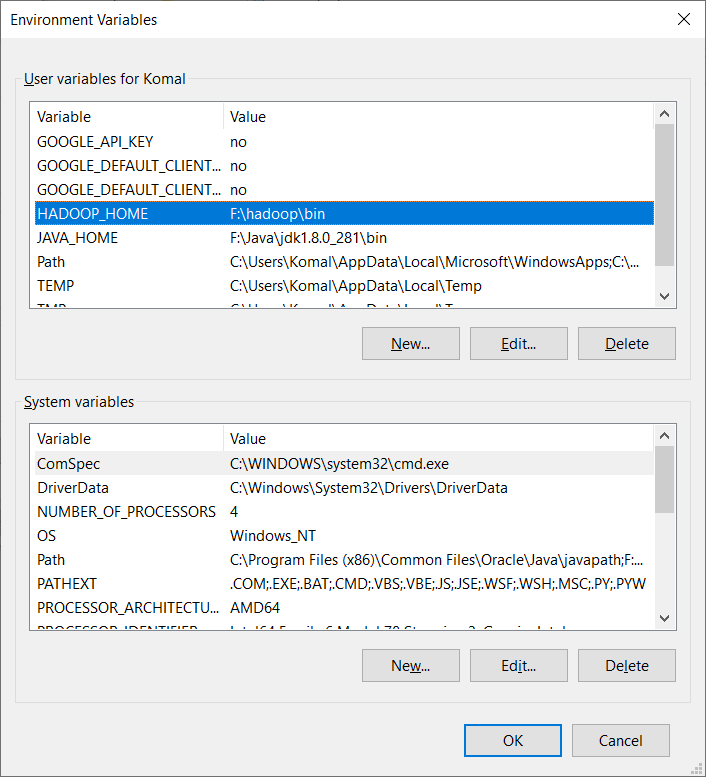


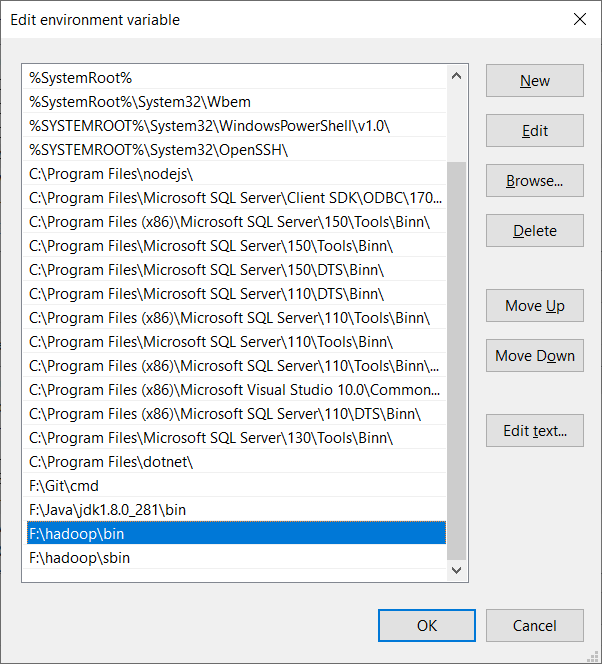
1. Now configure the **“hadoop-env.cmd”** file as below:



## Windows Configuration

1. Go to Windows => Settings => System and type environment in search and select “Edit the system environment variables” as shown in screenshot below. Now select “Environment Variables”. Refer section 3.2 a.
2. Now set the environment variable “**HADOOP\_HOME**” as well as path of **bin and sbin** folder in F:\hadoop. After setting click on Ok.

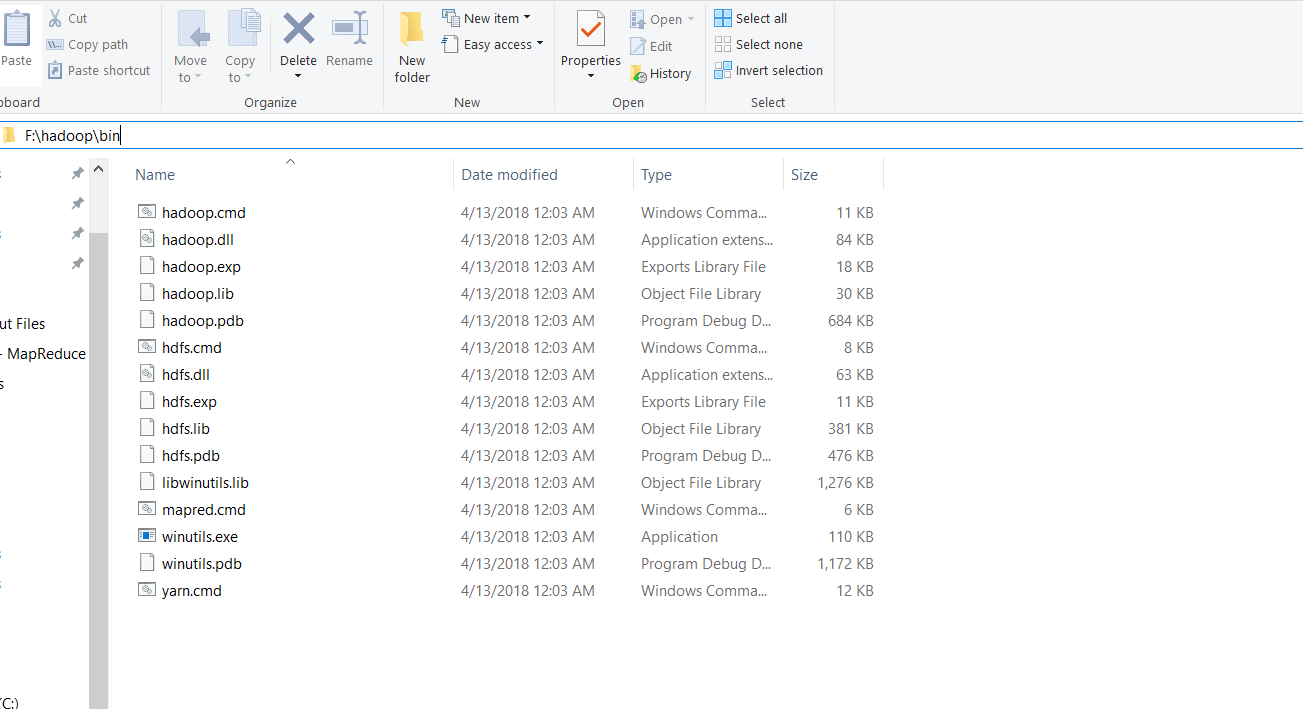




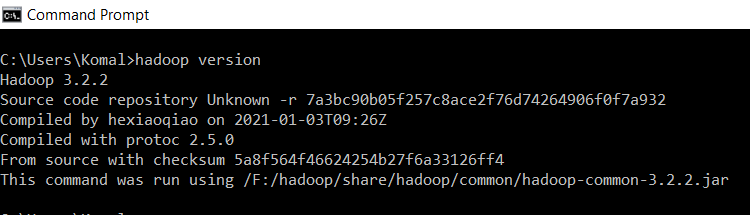
1. Now some configuration files are missing in folder **“F:\hadoop\bin”** so to fix thisdownload these configuration files from

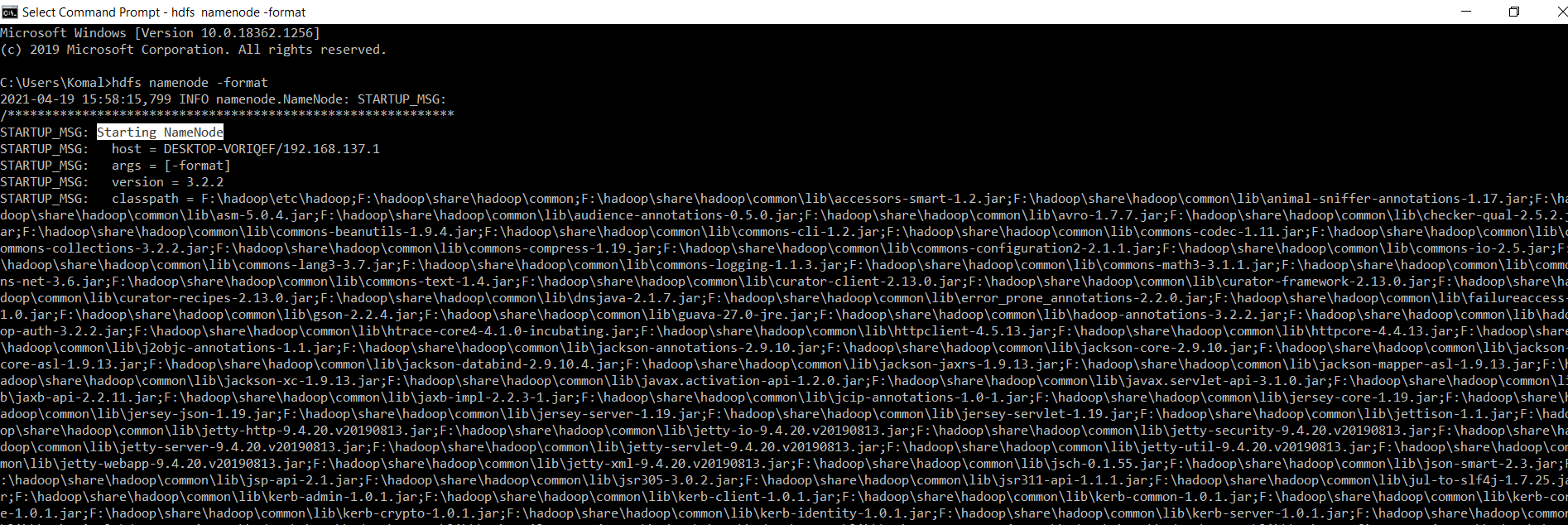
[**https://github.com/s911415/apache-hadoop-3.1.0-winutils**](https://github.com/s911415/apache-hadoop-3.1.0-winutils)

1. Now either rename the original bin folder inside “F:\hadoop” or delete it and move this new bin inside F:\hadoop\HadoopConfiguration-FIXbin\bin folder to “F:\hadoop”



1. Now Hadoop is successfully installed on our local system and to confirm it run the command **“hdfs namenode -format”** and start all the hadoop deamons in new command prompt using **“start-all.cmd”.**

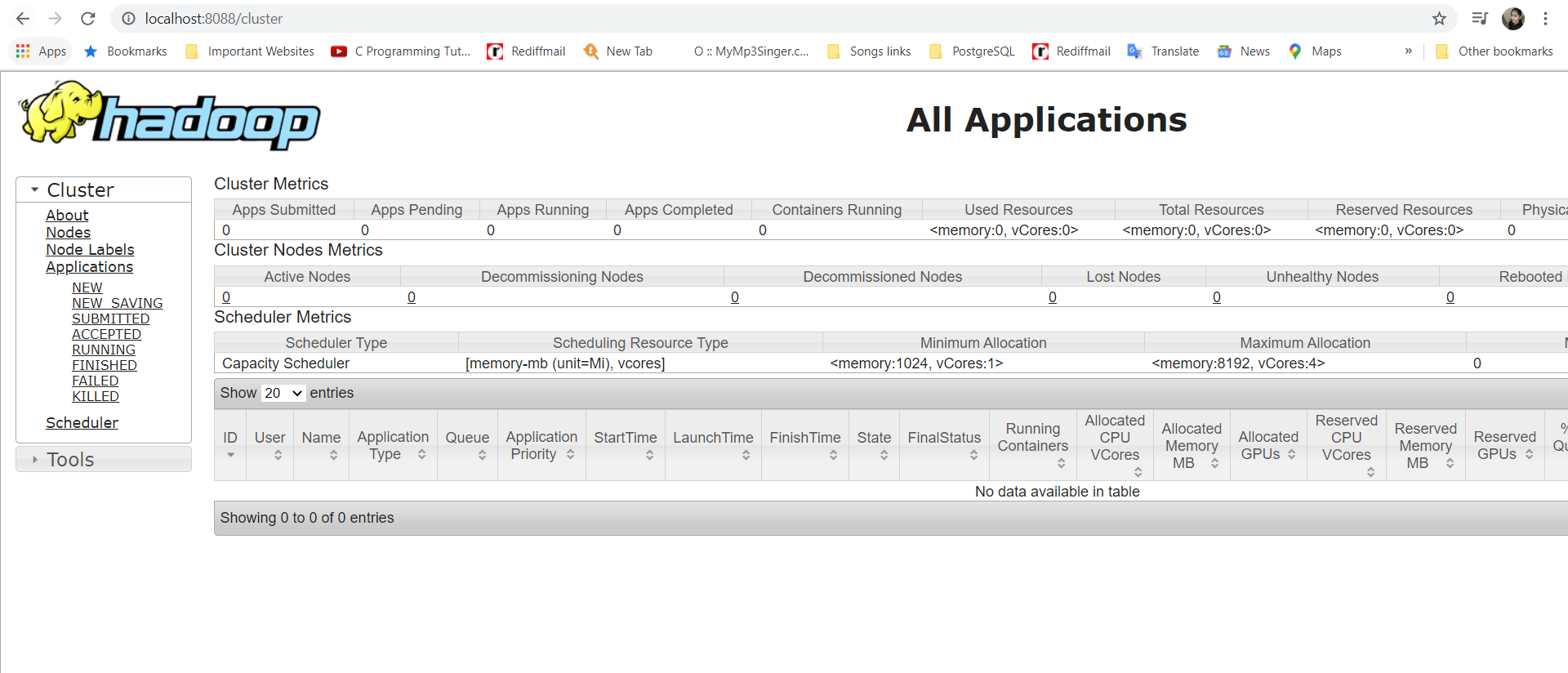




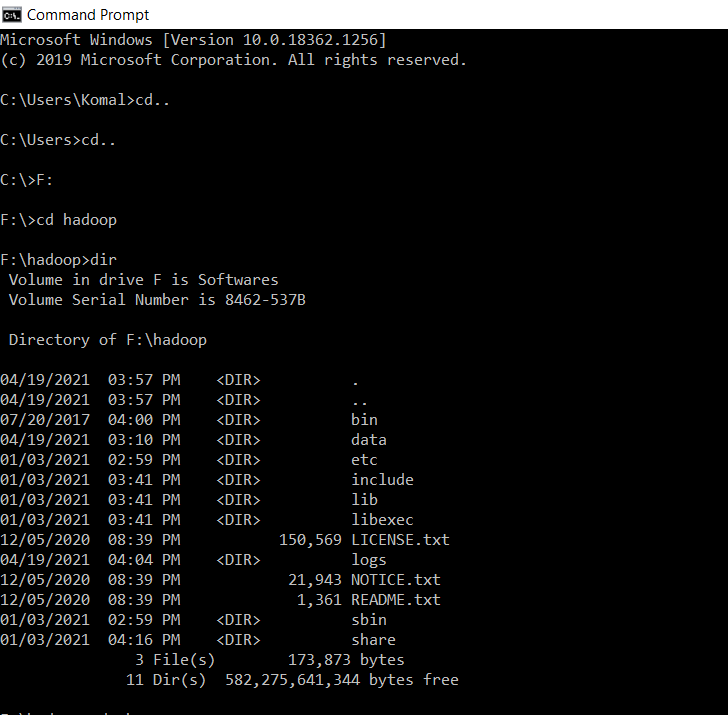


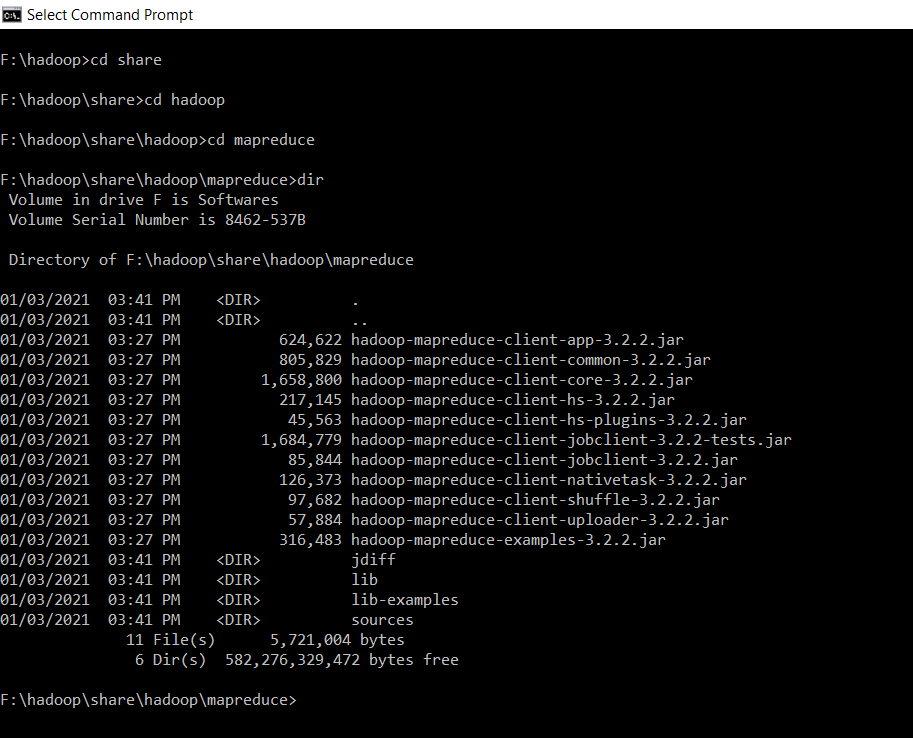
# Program Execution

1. Now verify that whether the daemons have started or not by accessing <http://localhost:8088/cluster> console.

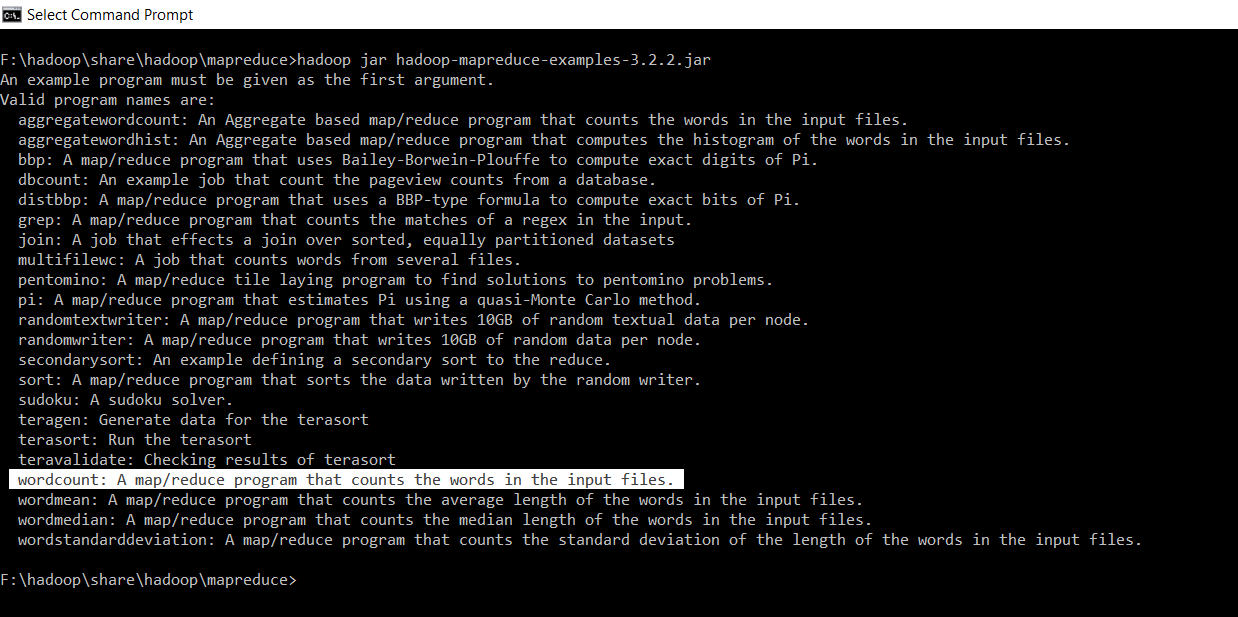


1. Word Count Program already comes with a mapreduce examples package in jar file so access the share folder inside which mapreduce folder is present

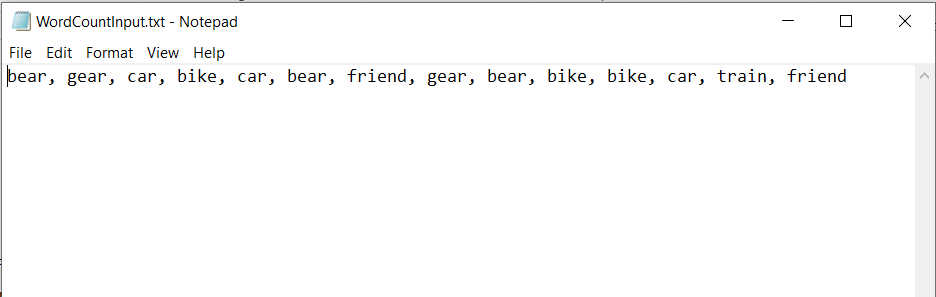




1. Check word count program is included in **“hadoop-mapreduce-examples-3.2.2.jar”** file

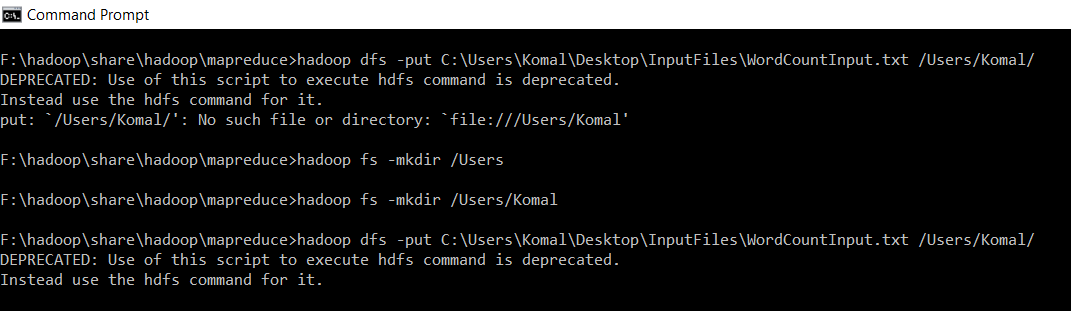


1. Create input file for which word count program to be executed and save it in any directory



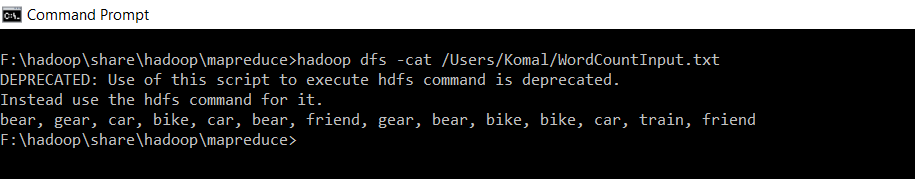
1. Move this input file to hadoop hdfs because mapreduce takes the input from hdfs and save the output in hdfs with the help of below command:’

**“ hadoop dfs -put C:\Users\Komal\Desktop\InputFiles\WordCountInput.txt /Users/Komal/ “**



1. Now check whether our file is successfully moved or not by command

**“ hadoop dfs -cat /Users/Komal/WordCountInput.txt “**

****

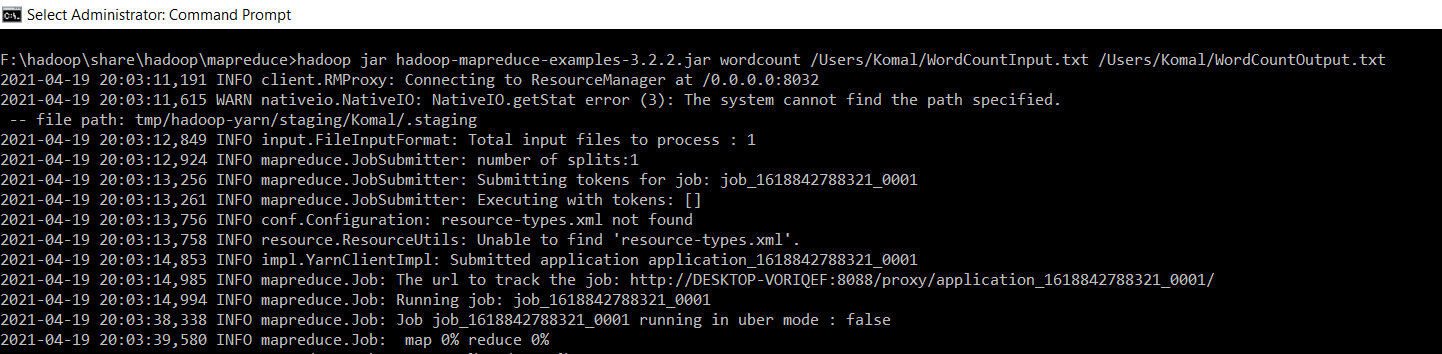
1. Open the new administrator command prompt.

Now execute the WordCount program with the help of below command

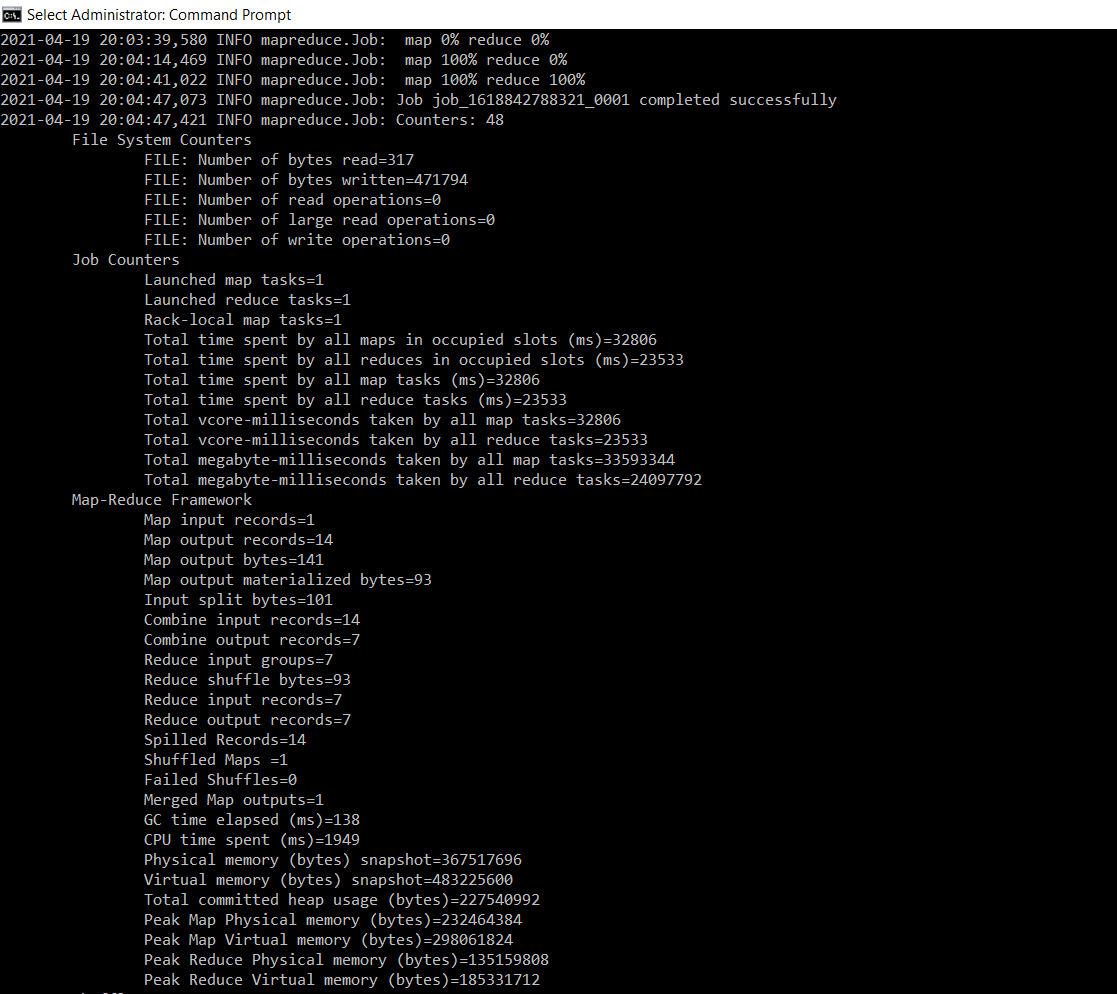
**“hadoop jar hadoop-mapreduce-examples-3.2.2.jar wordcount /Users/Komal/WordCountInput.txt /Users/Komal/WordCountOutput.txt”**

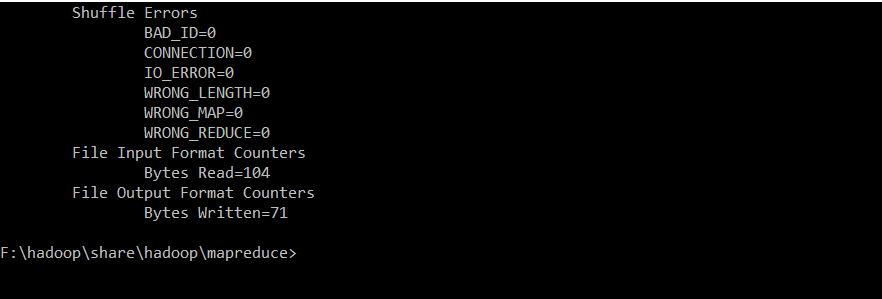
This command will execute the MapReduce algorithm for WordCount Problem in the package and produce the output file as “WordCountOutput.txt”

Intially both the Mapping and Reducing Job done are 0%

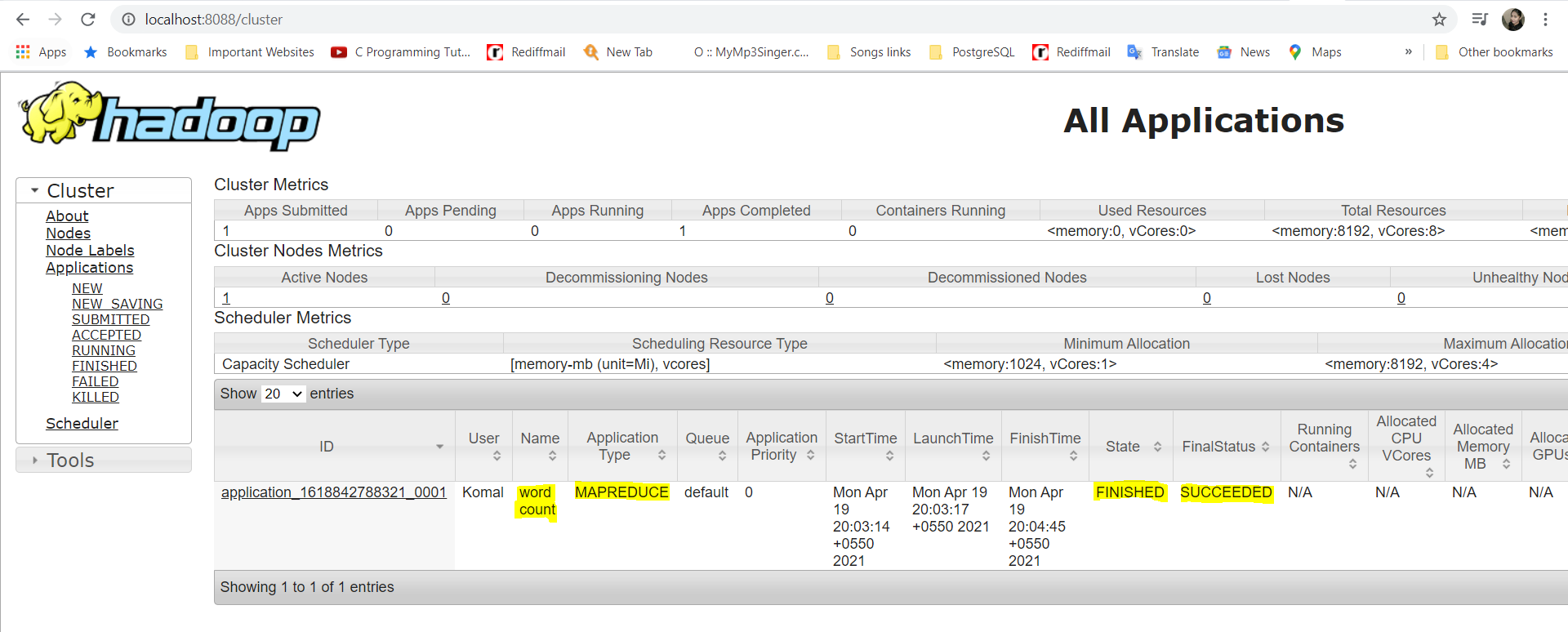
****

Then the program is started executing the Map & Reduce



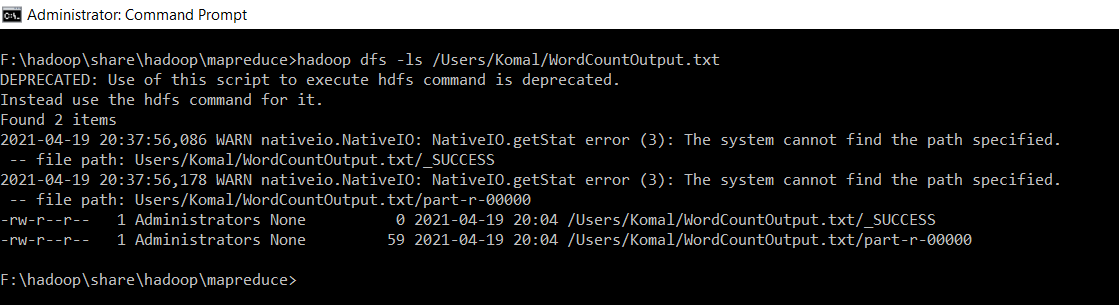


1. Also we can check on console whether the program is executed successfully or not as the state becomes FINISH and the status becomes SUCCEEDED on console <http://localhost:8088/cluster>



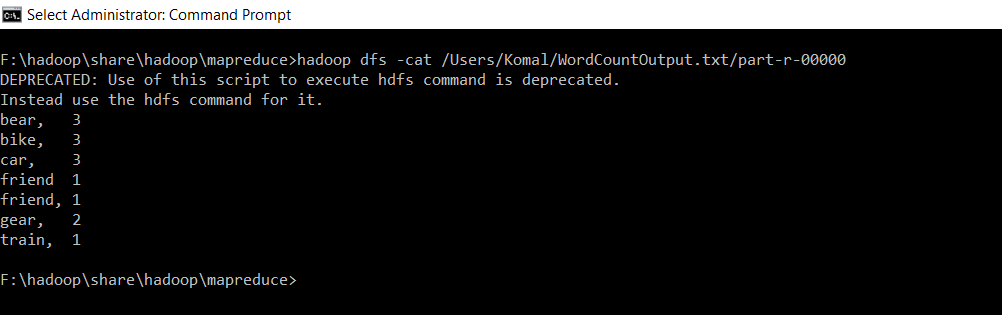
1. Now check whether output for WordCount program had successfully created or not with the help of below command:

**“hadoop dfs -ls /Users/Komal/WordCountOutput.txt”**



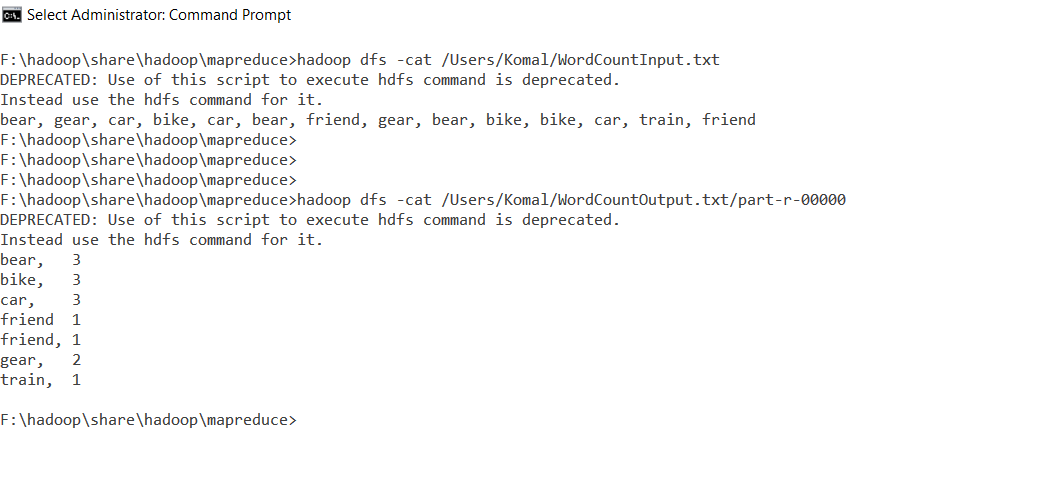
1. Now the output will be present in “/**Users/Komal/WordCountOutput.txt/part-r-00000”** so to see the final output, run below command

**“hadoop dfs -cat /Users/Komal/WordCountOutput.txt/part-r-00000”**



# Program Output

Output of the input file WordCountInput.txt is as below:



# 