**Big Data Analytics**

**Experiment No. 04**

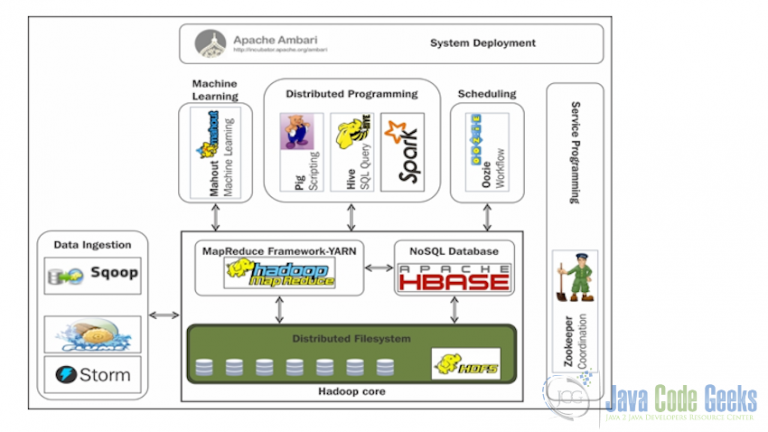
**Title: Hadoop Operations**

**Theory:**

* Hadoop is an open source software programming framework for storing a large amount of data and performing the computation.
* Hadoop is a framework that allow us to store and process large data sets in **parallel** and **distributed** fashion.
* Its framework is based on **Java programming** with some native code in C and shell scripts.
* Open Source, Distributed Processing, Distributed Storage, Reliable, Economical, Flexible, Fault-tolerant are the characteristics of the Hadoop.

**Hadoop Framework:**

There are *four major elements of Hadoop* i.e. **HDFS, MapReduce, YARN, and Hadoop Common**. Most of the tools or solutions are used to supplement or support these major elements.



**Hadoop Common**: These are Java libraries and utilities required by

Other Hadoop modules. These libraries provide filesystem and OS

level abstractions and contains the necessary Java files and scripts

required to start Hadoop.

**Hadoop YARN**: Yet Another Resource Negotiator (YARN),

introduced in Hadoop 2, a general purpose scheduler and resource

manager.

• **Hadoop Distributed File System (HDFS™):** A distributed file system

that provides high-throughput access to application data.

• **Hadoop MapReduce**: This is YARN-based system for parallel

processing of large data sets.

**Installation and Configuration of Hadoop:**

1. Install or download Java 1.8.0

<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>

1. Download the latest stable release from the Apache Hadoop downloads websites

<https://hadoop.apache.org/docs/r2.8.0/>

1. Install Java JDK 1.8.0 in "C:\JAVA“
2. Extract file Hadoop 2.8.0.zip and place under "C:\Hadoop-2.8.0".
3. Set the path HADOOP\_HOME Environment variable on windows 10.
4. Set the path JAVA\_HOME Environment variable on windows 10.
5. Next we set the Hadoop bin directory path and JAVA bin directory path.

Configuration:

1. Edit file C:/Hadoop-2.8.0/etc/hadoop/core-site.xml, paste below xml paragraph and save this file.

*<configuration>*

<property>

<name>fs.defaultFS</name>

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

</property>

*</configuration>*

1. 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>*

1. Create folder "data" under "C:\Hadoop-2.8.0". Create folder "datanode" under "C:\Hadoop-2.8.0\data". Create folder "namenode" under "C:\Hadoop-2.8.0\data" data.
2. Edit file C:\Hadoop-2.8.0/etc/hadoop/hdfs-site.xml, paste below xml paragraph and save this file.2.8.0/etc/hadoop/mapred-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>C:\hadoop-2.8.0\data\namenode</value>*

</property>

<property>

<name>dfs.datanode.data.dir</name>

*<value>C:\hadoop-2.8.0\data\datanode</value>*

</property>

*</configuration>*

1. Edit file C:/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>*

1. 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" (On C:\java this is path to file jdk.18.0).
2. Dowload file Hadoop Configuration.zip

(Link:https://raw.githubusercontent.com/MuhammadBilalYar/Hadoop-On-Window/master/Hadoop%20Configuration.zip).

1. Delete file bin on C:\Hadoop-2.8.0\bin, replaced by file bin on file just download (from Hadoop Configuration.zip).
2. Open cmd and typing command "hdfs namenode –format".

10. Open cmd and change directory to "C:\Hadoop-2.8.0\sbin" and type "start-all.cmd" to start apache.

11. Open: http://localhost:8088 to check if hadoop is running.

Commands of HDFS

(Before doing commands we want to this command so that it will easily started C:\Hadoop-2.8.0\sbin" and type "start-all.cmd" to start apache.)

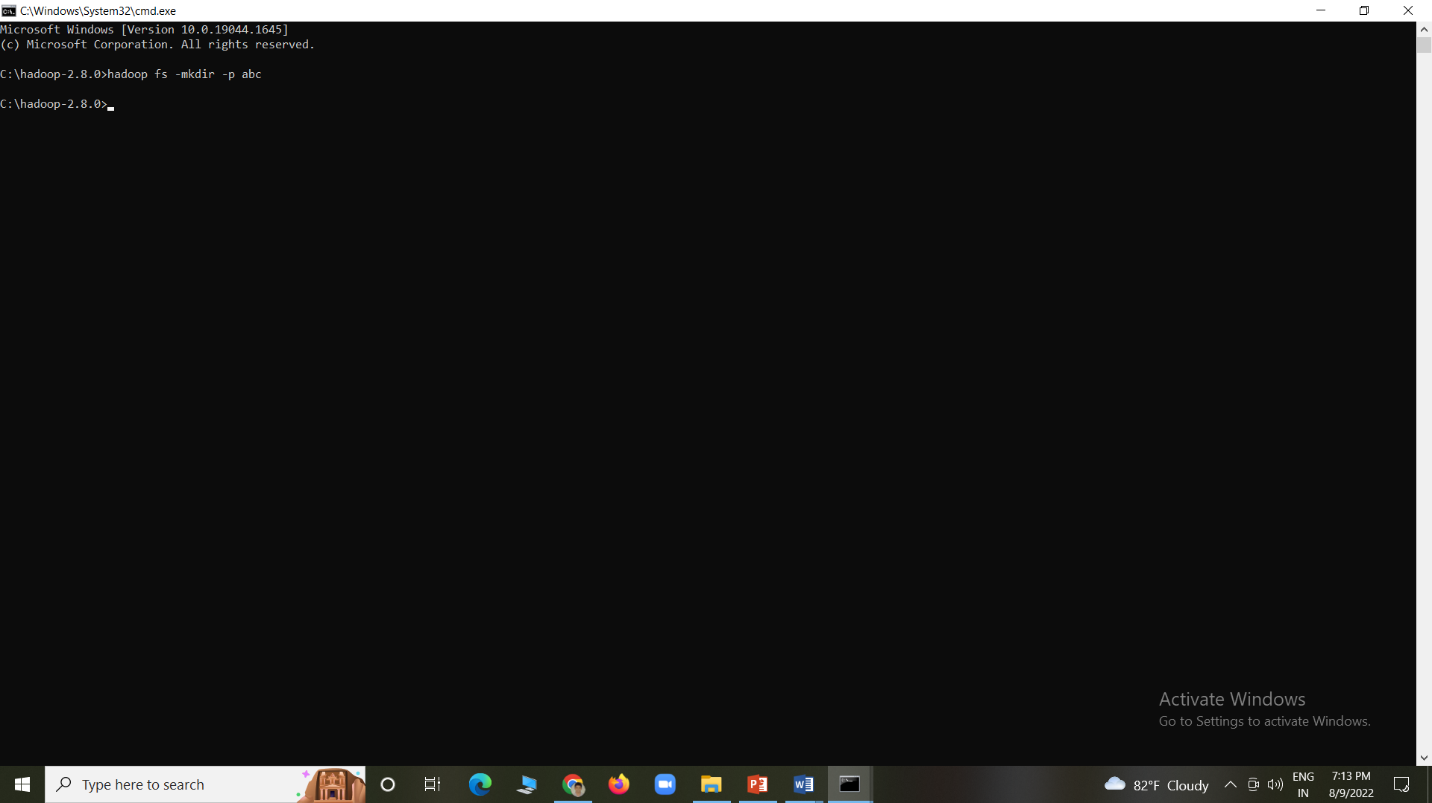
1. mkdir:

This is no different from the UNIX mkdir command and is used to create a directory on an HDFS environment.

–p mention not to fail if the directory already exists.

Syntax: hadoop fs -mkdir (path)

Example: hadoop fs -mkdir -p abc



1. ls:

This is no different from the UNIX ls command and it is used for listing the directories present under a specific directory in an HDFS system. The –lsr command may be used for the recursive listing of the directories and files under a specific folder.

–d The option is used to list the directories as plain files

–h The option is used to format the sizes of files into a human-readable manner than just number of bytes

–R The option is used to recursively list the contents of directories

Syntax: hadoop fs -ls (path)

Example:

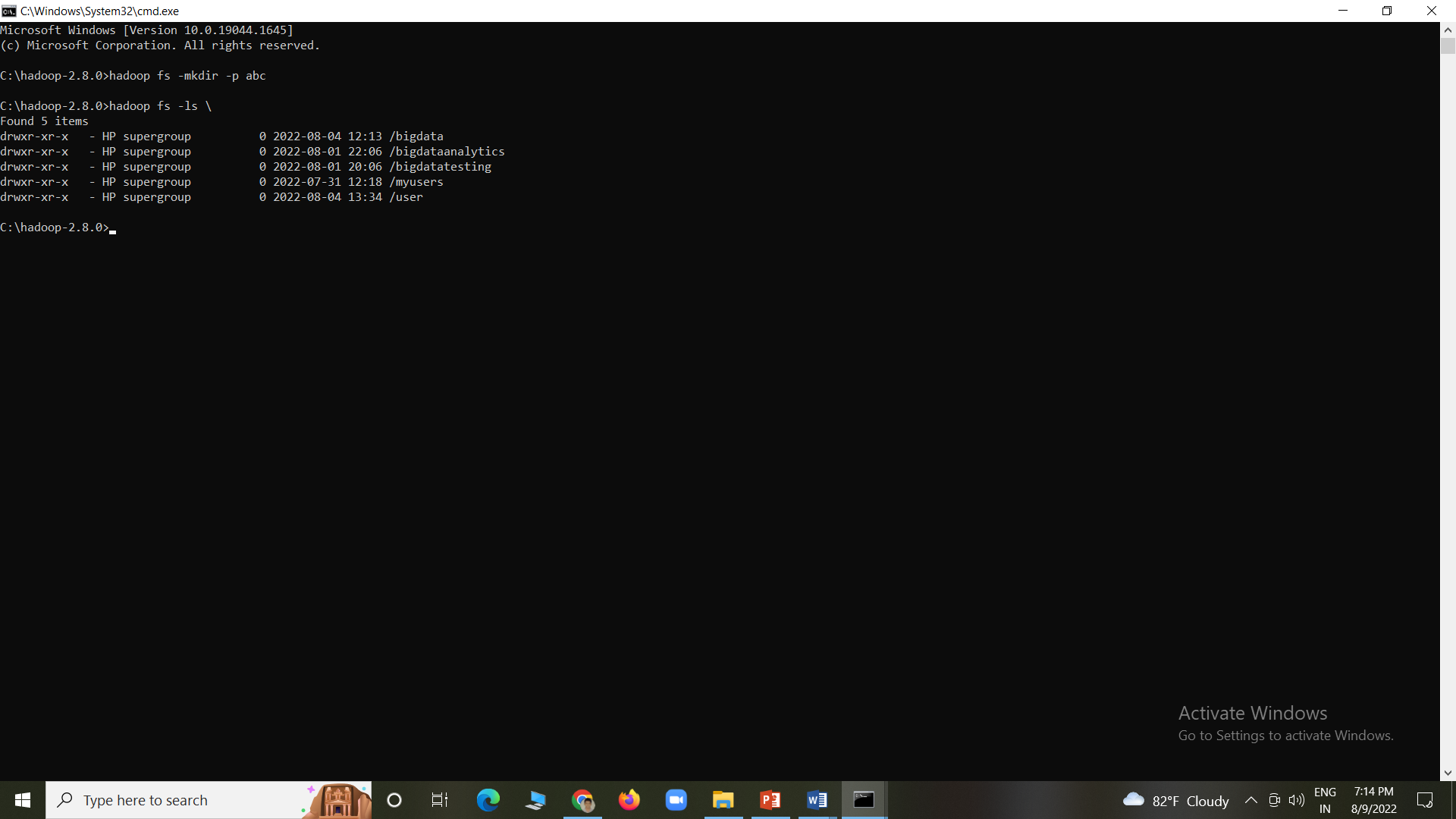
hadoop fs -ls \

Found 3 items

-rw-r--r-- 1 HP supergroup 122 2019-08-23 11:49 /desktop

drwxr-xr-x – HP supergroup 0 2019-08-27 11:03 /sample

drwxr-xr-x - HP supergroup 0 2019-08-27 11:03 /user



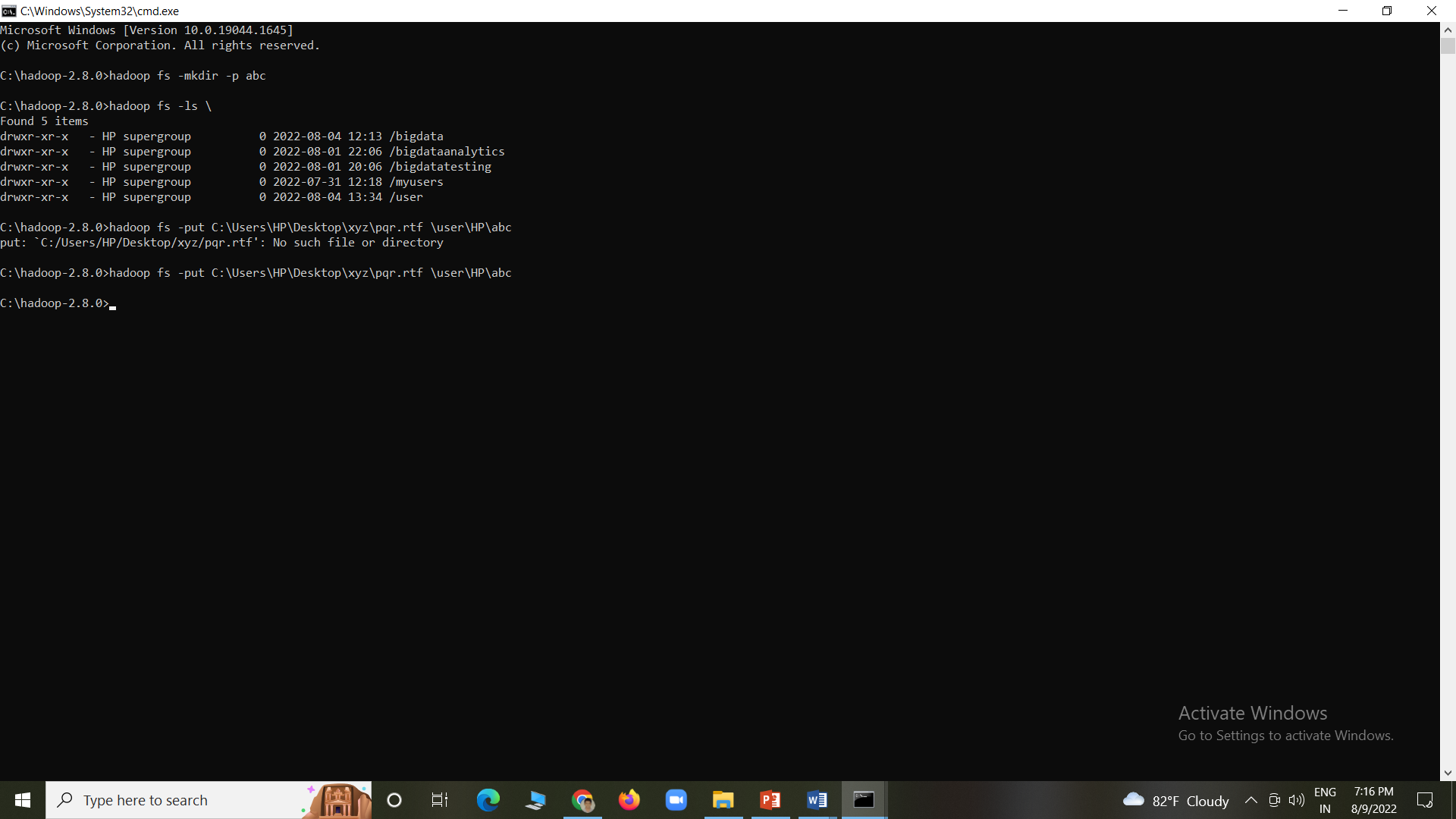
1. put:

This command is used to copy files from the local file system to the HDFS filesystem. This command is similar to –copyFromLocal command. This command will not work if the file already exists unless the –f flag is given to the command. This overwrites the destination if the file already exists before the copy

–p The flag preserves the access, modification time, ownership and the mode

Syntax: hadoop fs -put (source) (destination)

Example: hadoop fs -put C:\Users\HP\Desktop\xyz\pqr.rtf \user\HP\abc (Here we want to create a folder xyz in user-> desktop->xyz and then create an pqr,rtf file)

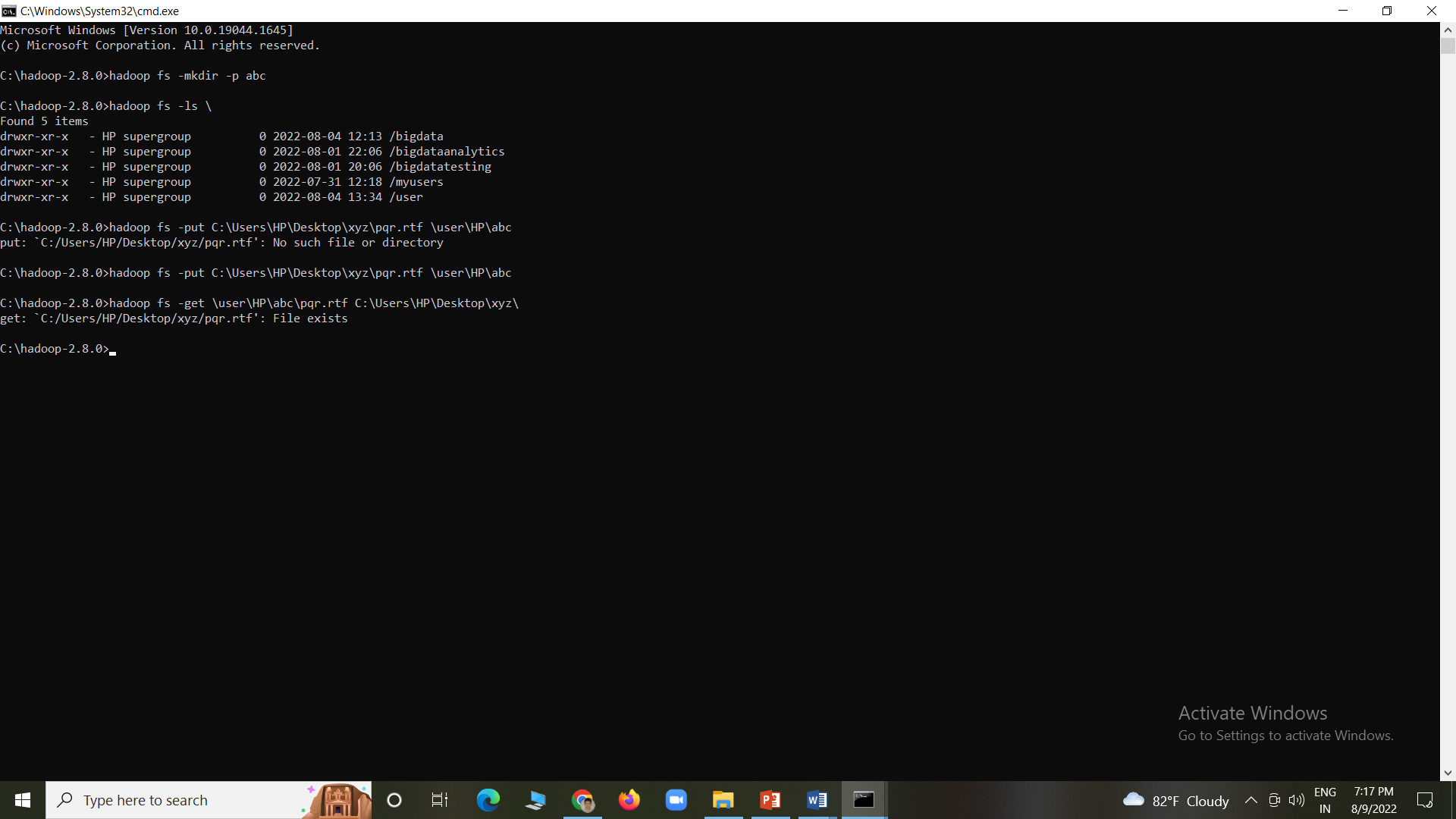


1. get:

This command is used to copy files from HDFS file system to the local file system, just the opposite to put command.

Syntax: hadoop fs -get (source) (destination)

Example: hadoop fs -get \user\HP\abc\pqr.rtf C:\Users\HP\Desktop\xyz\ (If file is there then the message comes file exists)



1. cat:

This command is similar to the UNIX cat command and is used for displaying the contents of a file on the console.

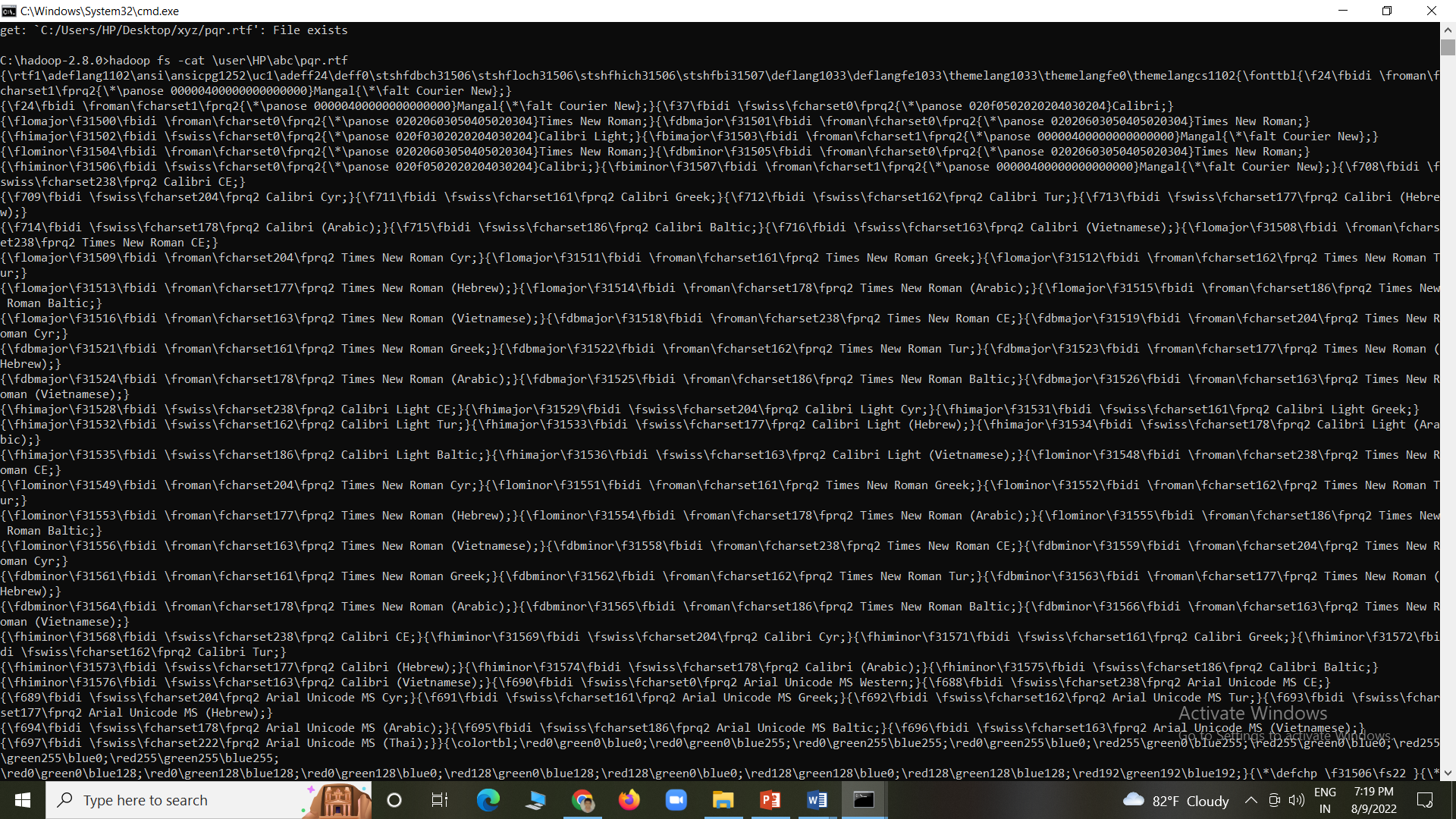
Syntax: hadoop fs -cat (path)

Example: hadoop fs -cat \user\HP\abc\pqr.rtf

{\rtf1\ansi\ansicpg1252\deff0\nouicompat\deflang1033{\fonttbl{\f0\fnil\fcharset0 Calibri;}}

{\\*\generator Riched20 10.0.17134}\viewkind4\uc1\pard\sa200\sl276\slmult1\f0\fs22\lang9 Hadoop\par

}

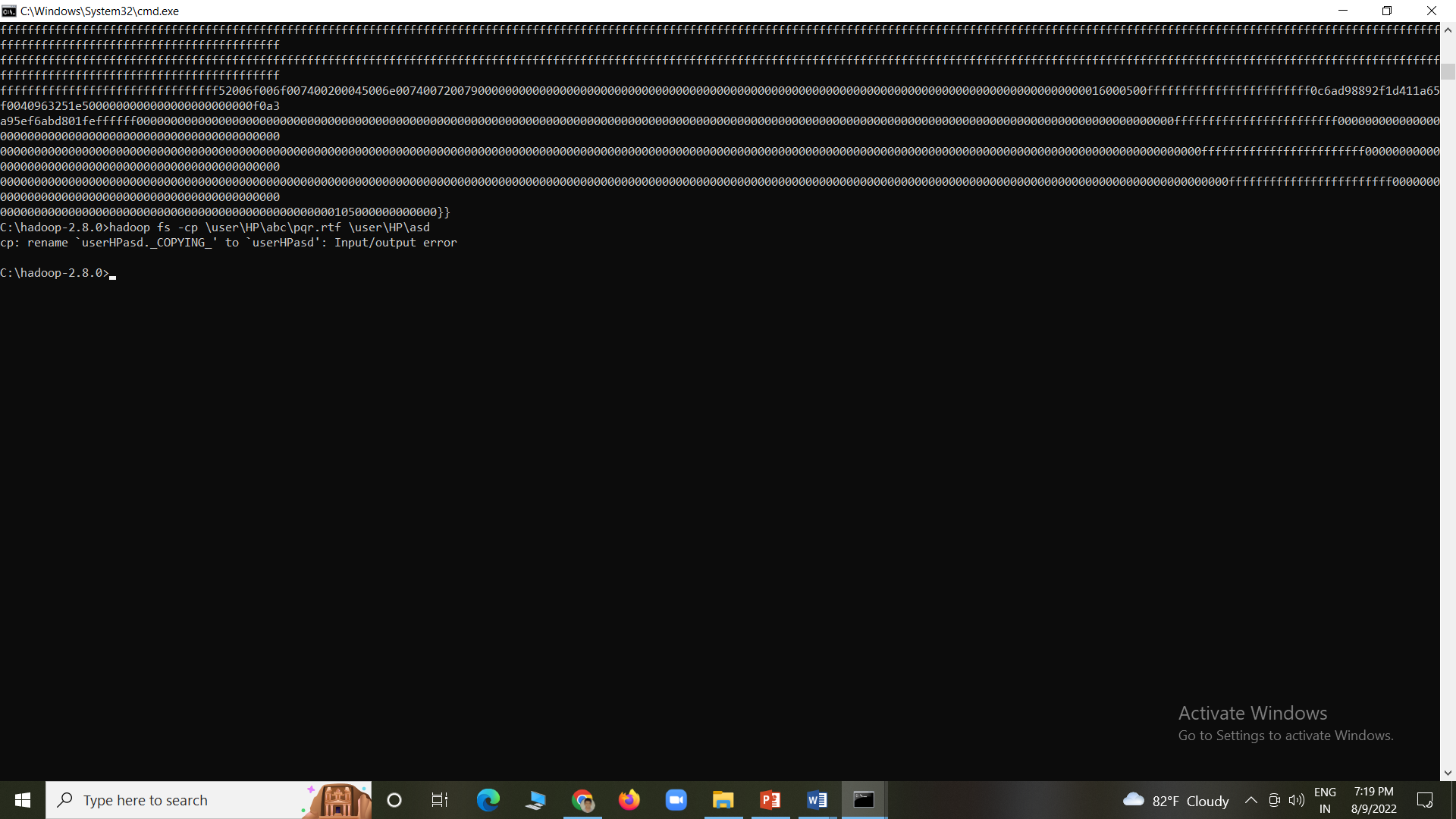


1. cp:

This command is similar to the UNIX cp command, and it is used for copying files from one directory to another directory within the HDFS file system.

Syntax: hadoop fs -cp (source) (destination)

Example: hadoop fs -cp \user\HP\abc\pqr.rtf \user\HP\asd

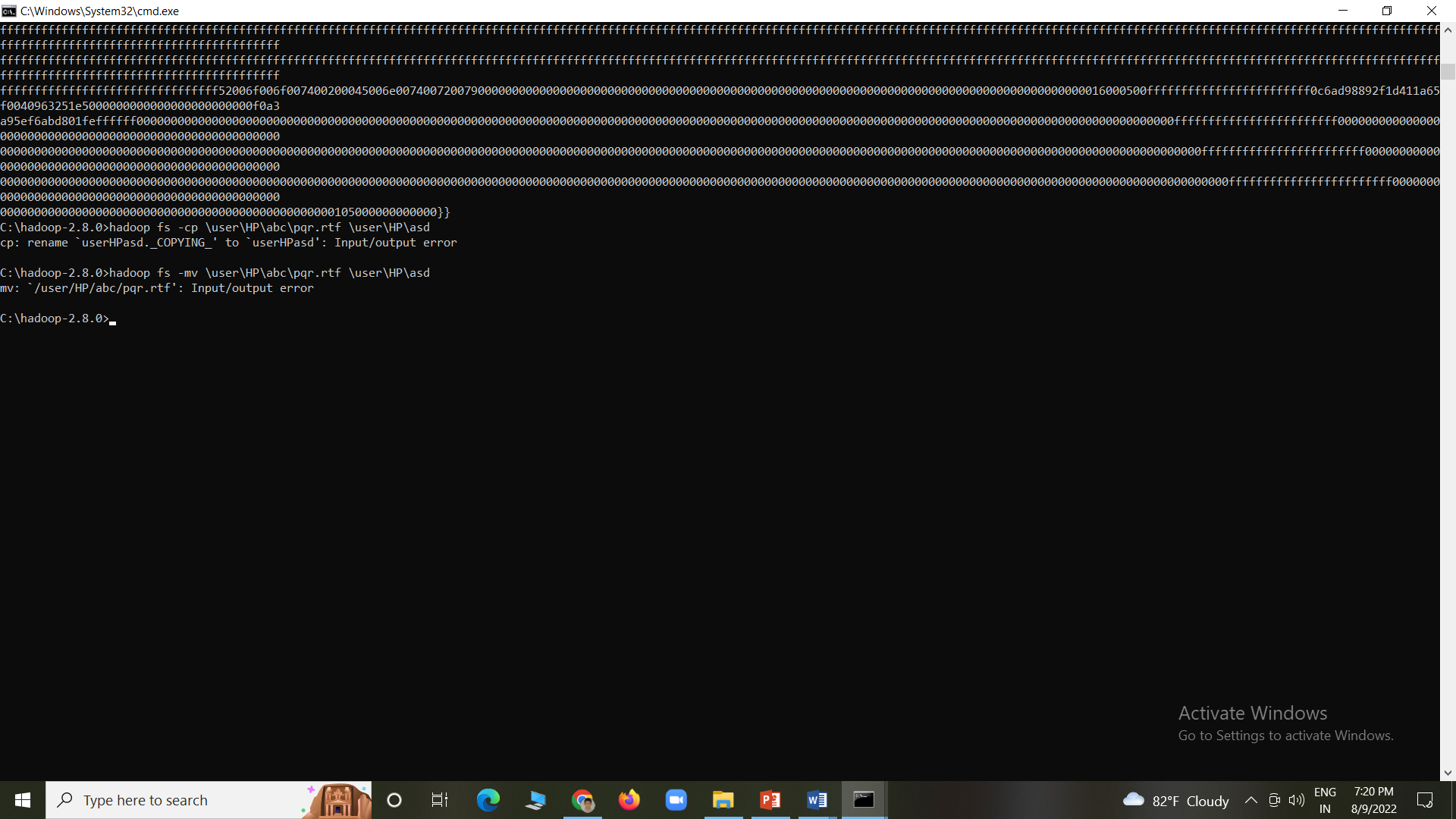


7. mv:

This command is similar to the UNIX mv command, and it is used for moving a file from one directory to another directory within the HDFS file system.

Syntax: hadoop fs -mv (source) (destination)

Example: hadoop fs -mv \user\HP\abc\pqr.rtf \user\HP\asd



1. rm:

This command is similar to the UNIX rm command, and it is used for removing a file from the HDFS file system. The command –rmr can be used to delete files recursively.

–rm Only files can be removed but directories can’t be deleted by this command

–rm r Recursively remove directories and files

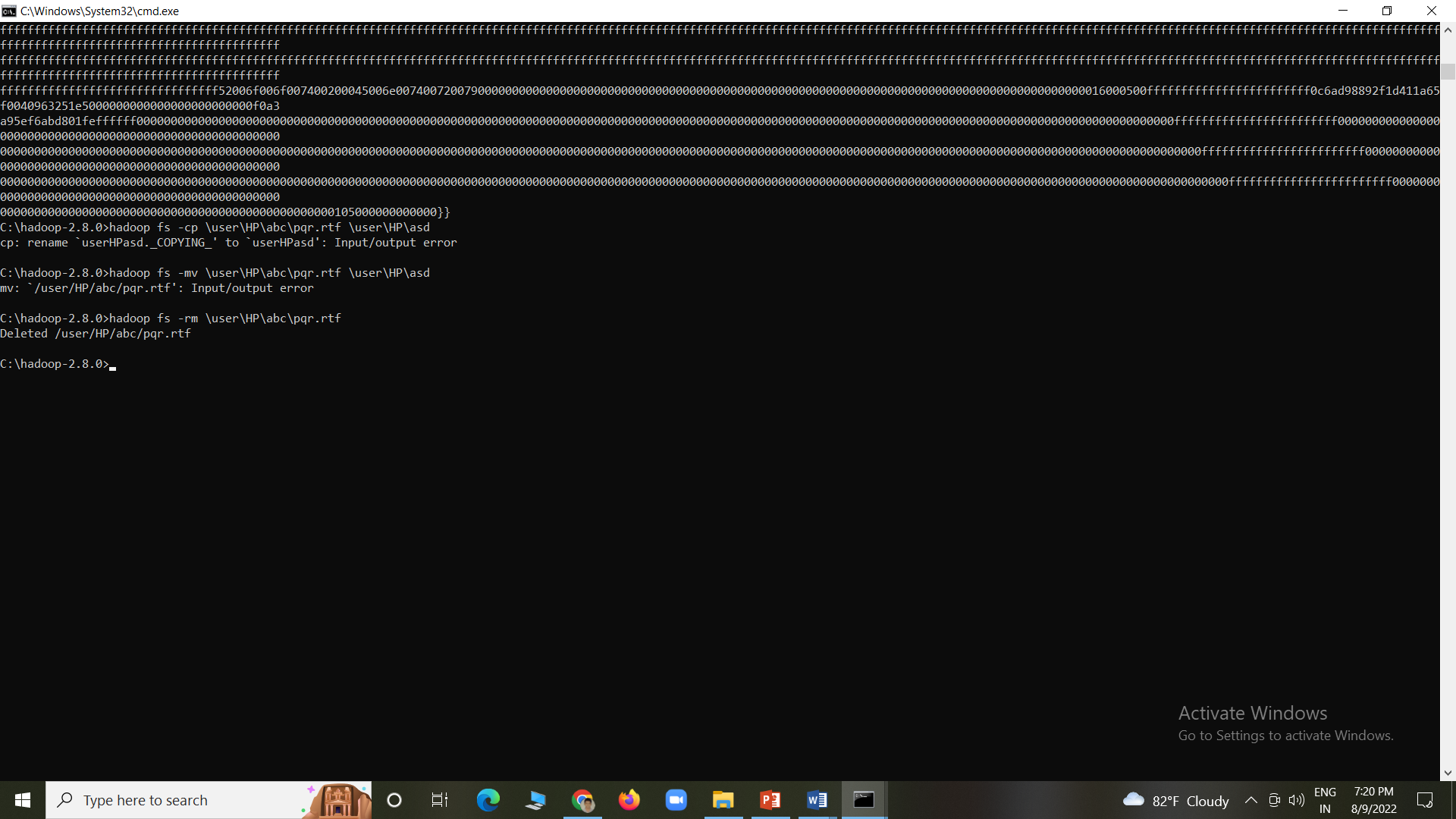
–skipTrash used to bypass the trash then it immediately deletes the source

–f mention that if there is no file existing

–rR used to recursively delete directories

Syntax: hadoop fs -rm (path)

Example: hadoop fs -rm \user\HP\abc\pqr.rtf



Conclusion :

We installed, configured Hadoop and HDFS commands in Hadoop.